UiO : Department of Informatics
University of Oslo

Time Dependent Virtual Machine Consolidation with SLA (Service Level Agreement) consideration

Yadassa H. Alla

Master's Thesis Spring 2015

Contents

List of Figures

List of Tables

Acronyms

VM/s: Virtual Machine/s
PM/s: Physical Machine/s
SLA: Service Level Agreement
QOS: Quality of Service
CPU:Centeral Processor Unit
CC:Cloud Computing
FFD: First Fit Decreasing

Acknowledgement

Above all, I would like to acknowledge God for all the supports behind all walks of my life.

Next,my special thanks goes to Oslo University informatics department Administration staffs for allowing me to study this masters program. Furthermore, I would like to thank Oslo and Akershus University college, computer engineering department lecturers, who meant a lot for the success. I owe thanks and respect for my supervisors Hugo Lewi Hammer (A.Professor at HioA) and Anis Yazidi(A.Professor at HioA) for their unreserved advises and supports on my work. My special gratitude will be for Hugo for his close follow up and guidance, patience and prompt when ever I am in need of help.

All my class mates deserve thanks for their companion. It makes my day when I read their posts on facebook. Specially, my appreciation goes to Diako Kezri for being a good friend during this two years.

Lastly, I am more than words thank-full to my families (my wife Marge, my daughter Kolu, my brother Jeto and his family)for their special attachment and upholding my psychology during my studies. Without them it wouldn't be possible to wind up these two years. My lovely mom, Amasu Disasa, deserves uncountable love, thanks, respect, gratitude and all special due for her daily prayers which means a lot for my life.

Abstract

Cloud data center is becoming the most essential infrastructure for computing services. In effect, the operational cost of a data center is also increasing drastically. To decrease this cost, consolidation of VMs with less degradation of performance is so important. To guarantee the expected Quality of Service (QOS) the important factors to be controlled are performance of the service including timely leverage and overall resource utilization of the data center. In this paper, we tried to investigate how to efficiently utilize resources with reduced SLA violation in a data center. In order to optimize efficiency, VMs ought to be consolidated as tight as possible. To achieve this, an algorithm based on first fit decreasing (FFD) bin packing is designed and implemented. Hence, the algorithm is implemented on the following three approaches to pursue the goal: a)Deterministic Approach, which is mainly based on mean of the individual VMs;b)Stochastic Approach I, which is basically done by treating individual VMs based on their mean and variances and ;c) Stochastic Approach II, which depends on mean and covariance of individual VMs. The results obtained show that consolidating VMs based on mean and variance(stochastic approach I) performed better than the other two approaches for minimizing total percentage of SLA violation and stochastic approach II performed better than the two approaches for minimizing the number of PMs in consolidation.

Chapter 1

Introduction

1.1 Motivation

Cloud Computing (CC) has drastically changed information technology (IT) since its emergence. The operating cost of Cloud data centers in the world is also increasing significantly as the technology advances. As described by Jonathan G Koomey [22], it is assumed that electricity used by data centers worldwide increased by about 56% from 2005 to 2010. In 2010, it was accounted to be between 1.1% and 1.5% of total electricity use respectively in the world. According to Natural Resource Defence Council (NRDC) study, in 2013, electric consumption of data centers in US was estimated to be 91 billion KW-hours, and projected to be 140 billion KW-hours in 2020. This costs 13 billion dollar annual expense for data centers and 100 million metric tones of carbon emission annually[16]. In the increasing world of data center operational costs, system administrator can play a significant role. Hence, it is duty of system administrators to find ways to reduce the overall operational cost of the data centers and optimize the resource utilization.

1.2 Overview

The rapid increase in Cloud Computing and IT end user focus has driven a big increase in Cloud data center importance. Because of this, data center administrators make constant effort to find ways to improve performance, enhance infrastructure density, and increase multi-tenancy capabilities. As more converged systems make their way into the data center, infrastructure optimization will be critical to maintain a high level of service[20]. Virtualization is one of the many ways of optimizing a data center.

In virtualization, Virtual machines (here after VM/VMs) are the most important components (resources). They depend on some other physical machine (Servers)or host to share resources such as CPU, memory,bandwidth and so on. These resources are consumed differently by each VM. Hence, it is customary to be interested in the relationship between the VMs based on their consumption of the physical machine´s

(host´s) resources, still relationship between them can also be studied using different methodologies. Correlating peaks and valleys of the resource consumption is one of them. Besides, applications running on each VM differ from one another, but still based on the communication between VMs, one can reach to a conclusion about the kind of relationship they have. Such relationships between VMs are discussed in detail by Xiaodong et al [34].

Studying the relationships among VMs helps to consolidate the VMs in a manner that can save resource utilization. One of the recommendations that the authors of [16] proposed to mitigate the power consumption problem of the data centers is to adopt server utilization metrics, like average utilization of the Central Processing Unit (CPU) to adjust the consumption if it surpasses a certain level. Reduction in resource utilization and energy consumption can be achieved by dynamically consolidating VMs and applying live migration, transferring VMs between physical servers with minimum downtime and the likes.

In Cloud Computing, one of the main goal of consolidating VMs is to efficiently save the resource utilization and maintain the SLA (Service level Agreement) for users as much as possible [12][2][29]. Degradation in QOS (Quality of service) from the side of service provider means violation of SLA, thus Cloud service providers always strive to meet the SLA they have with customers. As a result, Cloud service providers allow extra resource to maintain SLA. Hence, resulting in the server sprawl, over-provisioning of resources than the requirement of the workload. Similarly, consumers may also not use the resources they pay for due to reasonable factors. This results in wastage of resource by letting them idle. It is estimated that the average server utilization in many data centers is between 10% and 15%[24]. This is wasteful because an idle server often consumes about 70% of its peak power, implying that servers at low utilization consume significantly more energy than some servers at high utilization. Thus, to prevent such wastage, it is advisable to consolidate them on fewer servers. Consolidation may result in cut for power consumption while degrading performance, which results in SLA violation. The important concepts: Over-provisioning,allocating more resources than required to meet SLA and under-provisioning,allocating less resource than required to save the consumption cost, are the other constant trade-offs in Cloud Computing environment. Live migration,migrating VMs if their resource consumption surpasses the capacity of resource supply of their host, helps to mitigate these two concepts.

The trade-off that arises between performance and power-cost in consolidating VMs is the crucial point in today's data-centers[2]. Since power consumption goes linearly with number of physical machines, it is important to focus on minimizing the number of physical machines. Thus, this thesis contributes and proposes a solution for efficiently consolidating VMs having in mind reduction in overall operational cost of a data center. In addition reducing SLA violation will be considered as additional focus so that performance will not be shaken.

1.3 Problem statement

How to reduce operational cost of a data center by optimizing VM consolidation that reduces resource consumption and minimize violation of SLA by taking advantages of the dependence in usage patterns between VMs

Resource consumption in the problem statement is mainly related to consumption of physical machines. Reduction of physical machines in effect reduces electric consumption. Data center's consumption of electricity is increasing as technology grows. Therefore, the main goal of this thesis is to explore how to reduce the consumption of physical machines and as result reduce electricity consumption in a data center.

Optimizing refers to a technique that helps to achieve the most advantageous outcome in consolidating VMs. In this thesis, it is directly attached to the algorithm that helps in packing (consolidating) the VMs in to less number of physical machines and help efficiently utilize resources.

Dependence in Usage pattern refers to the way VMs are matched to be consolidated. VM with high load will depend on VM with low load to save resource consumption.

VM consolidation in the problem statement refers to packing together VMs based on their CPU utilization. Though there are many ways to consolidate VMs three methodologies are preferred.These are:- a)based on the mean CPU consumption of the individual VMs; b)based on mean and variance in CPU consumption of individual VMs; c) based on mean and covariance in CPU consumption of individual VMs.

SLA (Service Level Agreement) refers to the agreement between consumer or user and service provider to maintain agreed up on QOS (Quality of service). It is basically setting threshold for consolidating VMs. If for example the aggregate CPU utilization of VMs exceeds the capacity of the physical machine then migrating the VM to another physical machine and or take another action. Generally, SLA violation happens when the aggregate demand for the CPU performance of VMs surpasses the available CPU capacity of a physical machine (hosting server).

1.4 Challenges of VM consolidation

Several studies [1][13] have shown that the basic challenges for efficient and dynamic VM consolidation are as follows:

- Host underload detection: Deciding if a host is considered to be underloaded so that all VMs should be migrated from it and the host should be switched to a low-power mode (to minimize the number of active physical servers).

- Host overload detection: Deciding if a host is considered to be overloaded so that some VMs should be migrated from it to other active or reactivated hosts (to avoid violating the QoS requirements).

- VM selection: Selecting VMs to migrate from overloaded host. That is which VM to migrate.
- VM live migration: Performing VM migration process with minimal service downtime and resource consumption during migration process.
- VM placement: Where to migrate Virtual machines.

The above challenges will be used as an input to design our approach to consolidate VMs.

1.5 Approach

The problem statement of the thesis requires designing dynamic and energy efficient VM consolidation algorithm. Hence, according to the requirement of the thesis there are several ways to approach it. One of them is modeling and simulating the experimental setup using existing simulation tool such as ClousSim [1]. CloudSim is a toolkit (library) for simulating Cloud Computing scenarios. It provides basic classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system (eg. scheduling and provisioning) [15]. This approach would save time and money. Moreover, it is simple and can easily be reproduced than other experimental setup approaches.

Literature review is another alternative to approach the issue. But, conclusion deducted from literature might end up in biased outcome.

Thus, it is preferred to approach the issue using modeling and simulation based on real-workload traces obtained from PlanetLab collected by CoMon-project[2] logged randomly for 10 days in 2011.

Before starting with any other simulating process, the workload-traces obtained will be evaluated for further analysis to ease the research. Close relationship evaluation of the traces will be made based on the CPU utilization of each VMs. Mean, variance, covariance and correlation are some of the basic parameters to be evaluated for studying their relationship. More on how to approach the project comes under methodology in the corresponding chapter.

1.6 The goal of the thesis

The focus of this thesis is to design efficient and dynamic VM consolidation algorithm to reduce the cost of resource consumption in data centers. The key idea behind VM consolidation is to reduce the number of utilized physical servers and shutdown idle servers. This is basically achieved by consolidating the VMs as efficiently as possible. The important point to be

[1] http://www.Cloudbus.org/Cloudsim/

[2] https://github.com/beloglazov/planetlab-workload-traces

noticed when consolidating VMs is maintaining the SLA a service provider has with its customers. Since the designed algorithm is implemented in IaaS (Infrastructure as a Service), it is of great value to consider SLA. Overall goal of consolidating VMs is achieved by designing an algorithms that finds ready physical servers to allocate and migrate the VMs efficiently with minimum violation of SLA (Service Level Agreement).

1.7 The structure of the thesis

The rest of the thesis is organized as follows.

- *Chapter 2*: Deals with background of cloud computing data centers, virtualization, different technologies, consolidation techniques and related works
- *Chapter 3*: Introduces the architecture, setup and methodologies of the experiment.
- *Chapter 4*: Deals with result obtained from the application of the methodologies
- *Chapter 5*: Outlines and discusses the results obtained
- *Chapter 6*: Describes the conclusion and future work.

Chapter 2

Background

2.1 Background

In this chapter, all related technologies and related works will be discussed. Moreover, detail explanation of how other works are related to our work will be discussed here. Thus, understanding the following concepts and technologies will help to have concept of consolidating VMs.

2.1.1 Cloud computing

The emergence of cloud computing is rooted back in 1960s, and it keeps on developing since then. The term "cloud computing" came from two terms "cloud" which means accessing application as a service from anywhere in the world and "Computing" which refers to the services given by computing service providers such as Google,IBM, Amazon and Microsoft [14]. Moreover [18] defined clould computing as *"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. network, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"*.

The first important beginning time of cloud computing is the development of salesforce.com[1] in 1999 with the idea of delivering enterprise application via a website. The next development was the emergence of Amazon Web service in 2002 which gave a cloud-based services consisting computation, storage and many more. In 2006 Amazon launched its Elastic Computing Cloud (EC2) as a commercial web service that allows individual and enterprises (companies) to rent a computer on which they can run their own application.

[1] http://www.salesforce.com/eu/

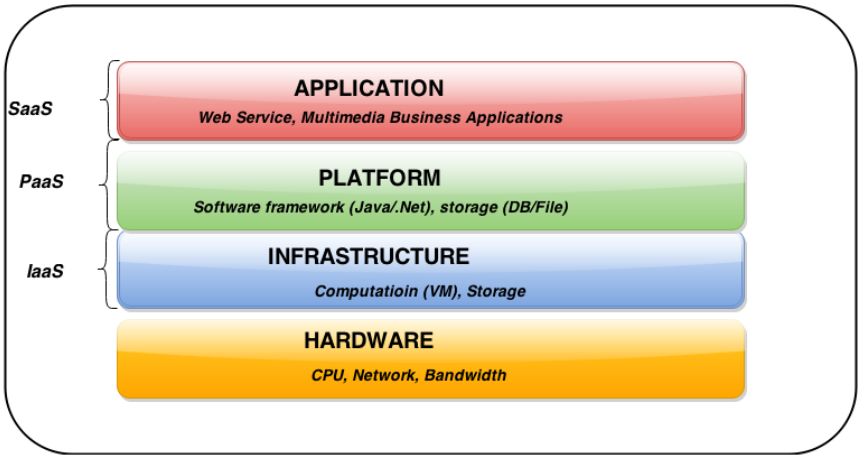


Figure 2.1: Service Layer

Recently cloud computing is giving computational services such as ***Software as a service (SaaS)***, a cloud computing model which gives service as a software or applications via network, typically the Internet by service providers for example Goggle Apps, Microsoft Office 365; ***Platform as a service (PaaS)***, is a model in which running an application becomes possible even with out having a hardware to run it. For example AWS Elastic Beanstalk, Heroku, Force.com, Google App Engine, Apache Stratos ; and ***Infrastructure as a service (IaaS)***, is a service model in which computational resources provided is of virtual type. For example Amazon EC2, Windows Azure, Rackspace, Google Compute Engine. Thus, customers can build their own platform on the provided virtual infrastructure[17]. Managing these services and their infrastructures become important in all IT arena. In addition to the service models mentioned above, [18] described the characteristics of cloud computing as follows:

- On-Demand self-service- a consumers have the capability of computing with intervention of service providers.
- Broad network access- The use of mobile phones, tablets, laptops and workstations through standard mechanism over the network.
- Resource pooling- service provider's resources are pooled by several consumers based on their demands. The location of these resources are not known by consumers.
- Rapid elasticity- Scalability for provisioning are almost unlimited.
- Measured service- cloud systems control and optimize resource usage.

As they discussed further cloud computing can be deployed in one of the following ways:

- Private clouds- such clouds are provisioned for exclusive use by organization having multiple users. It can be managed and controlled by an organization or providers.

- Community cloud- such clouds are used by a group of people, specific community of users have shared concerns.

- Public cloud-provisioned to general public such as business, academic and the likes

- Hybrid cloud- are combination of the above mentioned clouds.

Thus cloud computing management emerged for the sake of managing cloud computing infrastructures and services. In-addition to Amazon EC2[2] other cloud management systems such as Euclaptus[3], CloudStack[4] and OpenStack[5] have emerged.

2.1.2 Virtualization

In computing technologies, virtualization is creating a virtual version rather than real (actual) version of devices such as hard disks, servers, network and even operating systems and many more[28]. There are three types of virtualization:

- **para virtualization**, in which a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains;

- **partial virtualization**, in which some but not all of the target environment is simulated; Some guest programs, therefore, may need modifications to run in this virtual environment; and

- **full virualization**, in which almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified[30].Without virtualization of data centers, it wouldn't have been possible to talk about virtual machines (VMs). One of the benefits of virtualization is VM conoslidation for efficient use of electricity and other data center resources.

[2]http://aws.amazon.com/ec2/
[3]https://www.eucalyptus.com/
[4]https://cloudstack.apache.org/
[5]https://www.openstack.org/

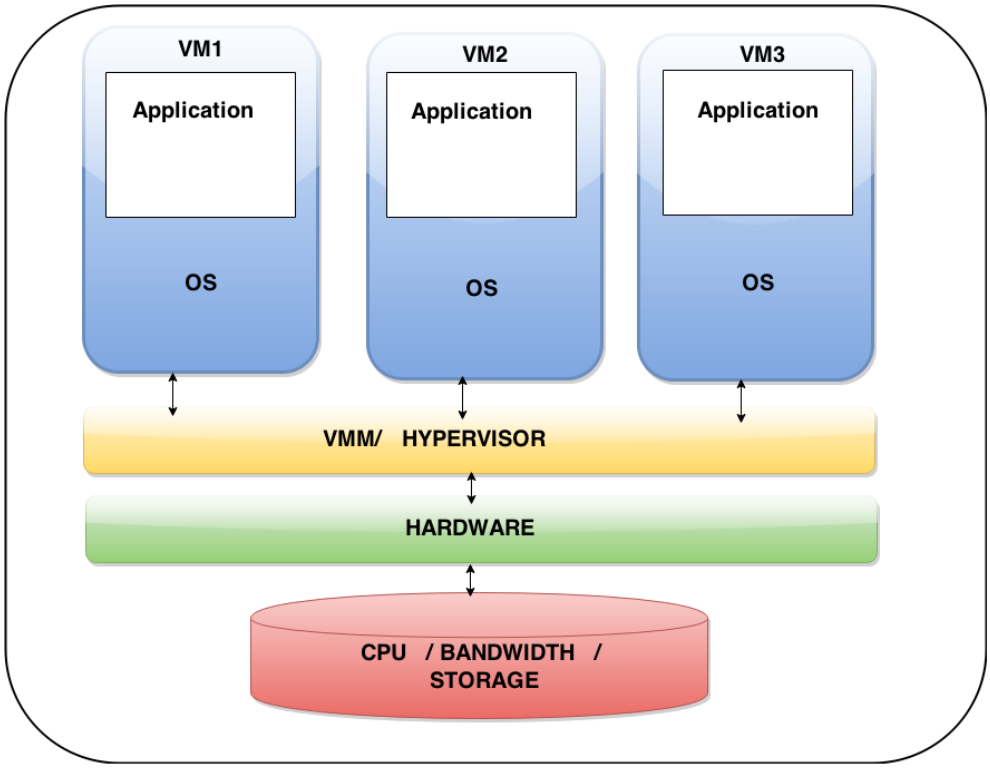


Figure 2.2: Virtualization Layer

2.1.3 Hypervisor

A hypervisor is a software, firmware or hardware that creates and runs virtual machines. It is also called virtual machine manager. Virtual machine manager collects resource usage information, such as CPU utilization, memory consumption and so on of a physical machine. There are two types of hypervisors:

- 1) **Naive or bare-metal hypervisors**, also called Type I Hypervisor, in this type of hypervisor software will be installed on the server as an operating systems and resources are paravirtualized and forwarded to running virtual machines ; and

- 2) **Hosted hypervisors**, also called Type II hypervisor, in such type of hypervisors a software will be loaded on top of already running operating system, for example installing virtual box on top of windows 8 or windows server 2008[21].

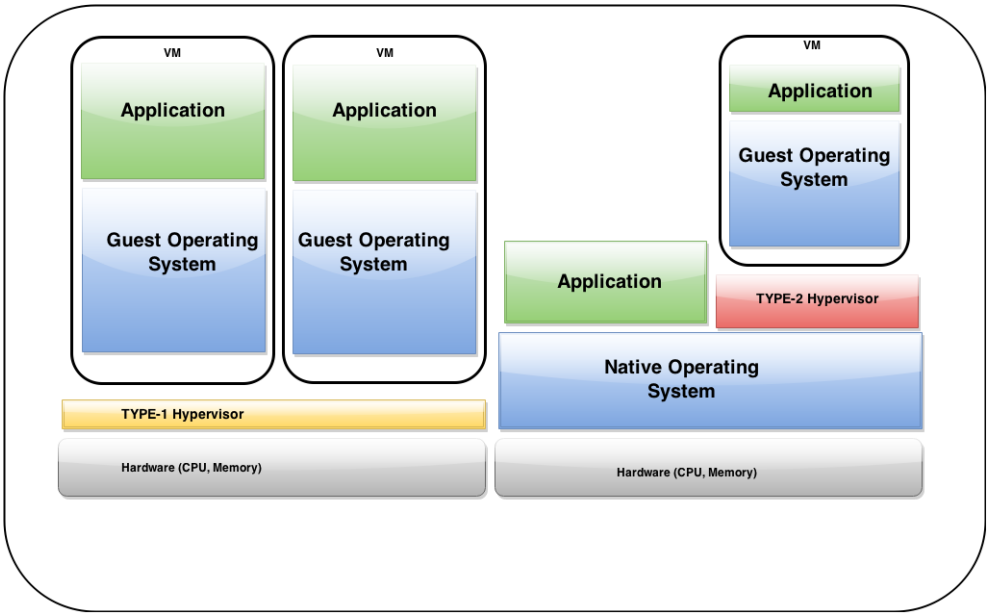


Figure 2.3: Type I and Type II hypervisors

2.1.4 Virtual Machines

A virtual machine is a program computer, that acts like a physical computer. The functions of virtual machines depend on the functionality of the real physical machine they target. Thus they can either be system virtual machine in which they provide a total substitute of the target real machine or process virtual machine in which they are designed for computing specific computer program[32].

2.1.5 Virtual Machine (VM) Consolidation

In virtualization environment, consolidation of virtual machines is one of the techniques used to save the operational costs of data centers. Hence, virtual machine consolidation refers to the use of a physical server to accommodate more than one virtual machine for the efficient use of resources. Co-locating (consolidating) VMs reduces the number of physical servers and reduce server sprawl,a situation in which multiple, under-utilized servers take up more space and consume more resources than can be justified by their workload[25]. In addition, VM consolidation reduces the power consumption, since power consumption and the number of servers are directly related,see also sub section 2.1.8 . VM consolidation can be performed in three ways: **a)static**, in which the virtual machine monitor (hyperviser) allocates the resource (physical resource such as memory, CPU and the likes) once and VMs will stay for long time period (such as months and years) on one physical machine. That means there will be no adjustment to the variation of workloads; **b)Semi-static**, in which VMs are place based on daily or weekly bases ; **c) Dynamic**, by adjusting depending

on the workload characteristics (Peak and off-peak utilization of resources) and make adjustment in hours and needs run-time placement algorithms . Dynamic VM consolidation helps in the efficient use of data centers [6] [26] [14][7]. In order to consolidate VMs there are several processes that must be undertaken. These are, vm selection, vm placement and vm migration.

VM Selection

VM selection is one of the challenges of VM consolidation process. It deals with migrating VMs until the physical machine is considered to be not overloaded. In VM selection process there are several policies to be followed for effective accomplishment of the process. These policies are discussed in detail by Heena Kaushar and et al.[3] as follows :a) The minimum Migration Time Policy; b) the maximum Correlation Plicy; c) The Random Choice Plicy and; d)Highest Potential Growth (HPG). Moreover, it is also described in detail by [13][11] as follows:a)Local Regression;b)Inter-quartile Range;c) Median Absolute deviation.

VM Placement

The process of selecting the most suitable host for the virtual machine, when a virtual machine is deployed on a host, is known as virtual machine placement, or simply placement. During placement, hosts are rated based on the virtual machine's hardware and resource requirements and the anticipated usage of resources. Host ratings also take into consideration the placement goal: either resource maximization on individual hosts or load balancing among hosts. The administrator selects a host for the virtual machine based on the host ratings.[27]

VM Migration

Migration of VM's can be accomplished by two methods:offline migration, which has downtime because of suspend and resume of operation and; live migration, which is widely used in cloud computing and uses copying before migrating to avoid downtime.One of the most remarkable features of virtualization is Live migration of VMs. In live migration active VM is transfered from one physical machine to other keeping the current working status of a VM while running. Such actions are a de facto in KVM[6] and Xen[7]. According to [8][1] there are three types of live migration techniques.

- Fault Tolerant Migration Technique:This technique migrates the VMs even-if system failure occurs during migration. It was assumed to minimize performance degradation of applications and improve availability.

[6] www.linux-kvm.org/
[7] http://en.wikipedia.org/wiki/Xen

- Load Balancing Migration Technique: This technique distributes load across the physical servers to improve the scalability of physical servers. It helps in in minimizing the resource consumption, implementation of fail-over, enhancing scalability, avoiding bottlenecks and over provisioning of resources etc.
- Energy Efficient Migration Technique: The power consumption of Data center is mainly based on the utilization of the servers and their cooling systems. Thus, migration techniques that conserves the energy of servers by optimum resource utilization is of focus.

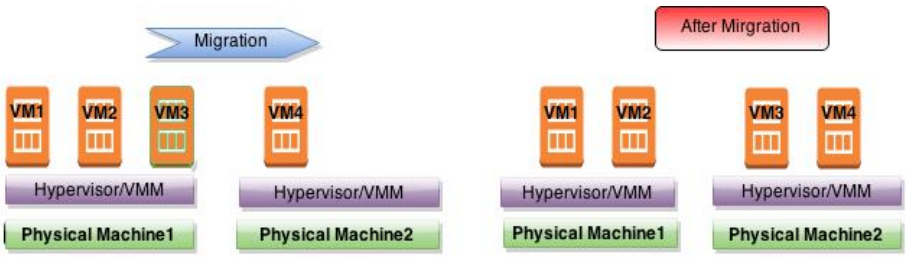


Figure 2.4: Live Migration of VMs

Benefits of VM Consolidation:

- Reduce operational costs:
  Hardware cost: Efficient use resources reduces the number of resources used when consolidation are applied.
  Power Cooling cost: the effect of reduction in the use of physical resources such as servers and racks leads to reduction in power and cooling system in a data center.

- Easy backup- thanks to snapshot taking back up is very easy.
- Deployment advantage- redeploying is easy by enabling snapshots.
- Green environment: reduction in use of electricity favours to green environment.
- Testing: Testing is very easy. Thanks to snapshot
- Server Sprawl-No over provisioning any more
- Decreased labour cost
- Reduced maintenance cost

Negatives of VM Consolidation

If proper management is not applied VM consolidation can end up in the following disadvantages.

- Overheads:Migration and placement of VMs can cause impact on performance of applications.This leads to overhead on network link and CPU utilization.
- Performance problem: VM consolidation can cause performance problem due to contention that arises from using the same physica resource such as CPU, memory and others.
- Single Point of failure: The main goal of VM consolidation is to pack as many VMs to a single physical server as possible. If necessary actions such as taking snapshots and others are not taken it can cause system failure.

Beside the above mentioned negatives, VM consolidation benefit outweighs and large data centers are using it increasingly.

2.1.6 Bin Packing

Bin packing problem is a combinatorial NP-hard problem, in which objects of different volumes must be packed into a finite number of bins in a way that minimizes the number of bins used[31].

Given a list of objects and their weights, and a collection of bins of fixed size, there are several heuristic method developed to pack the objects. These are:**Next Fit Heuristic (BP-NF)**, in which items are placed in the order of their arrival and the next item is placed in to the current bin if it fits, otherwise start a new bin; **First Fit Heuristic (BP-FF)**, in which items are placed according to their arrival but the next item will be placed into the lowest numbered bin in which it fits, otherwise start a new bin; **Best Fit Heuristic (BP-BF)** in which the next item will be placed into the bin which leave the least room left over after the item is placed in the bin, other wise start new bin; **Worst Fit Heuristic (BP-WF)**, in which the next item will be placed into the bin which will leave the most room left over after the item is placed in the bin, other wise start new bin; **First Fit Decreasing Heuristic**, sort the item in decreasing order and place the next item into the lowest numbered bin in which it fits, otherwise open a new bin; **Best Fit Decreasing Heuristic**, sort the item in decreasing order and place the next item in to the bin which will leave the least room left over after the item is place in the bin,otherwise start new bin. The most important and widely used approaches from the above methods are the last two, Best Fit Decreasing Heuristic and First Fit Decreasing Heuristic, approaches [9][31][33].

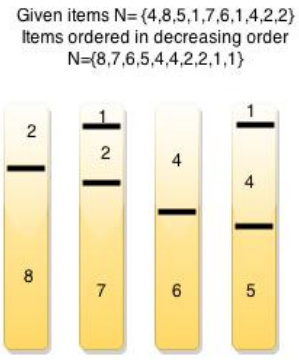


Figure 2.5: Bin Packing:First Fit Decreasing when large items are packed first

2.1.7 Service Level Agreement

A Service Level Agreement is an agreement between two or more parties, where one is user and the other is service provider[23]. As mentioned in chapter 1 Cloud Computing services, i.e IaaS,PaaS and SaaS, require an SLA because of the complex nature of cloud environment. The undesirable effect in VM consolidation is violation of service level agreement (SLA) for users. Service providers use over-provisioning which would result in wastage of resources if the users are not efficiently using the resource provided to them.One of the challenges of cloud computing data center is performance degradation due to violation in SLA. The peaks of a workload creates violation if resources are not adequately allocated. On the other hand if resources are allocated based on their peaks, it will leave hole for the wastage of extra resources. Therefore, cloud computing service providers introduced two types of scaling resources based on the demand[24]. These are, schedule based and rule based. Scheduled-based technique uses the daily cyclical workload pattern of a VM while rule-based basically depend on the rule if conditioin, if CPU utilization is greater than X for example. The rule-base can further be splited to reactive, that reacts to changes to a system, and proactive, which anticipate the future needs. The proactive rule-based technique uses parameters such as mean.

2.2 Auto-Scaling Techniques

Auto-scaling is a process of mapping resource demand to available resource in cloud computing. It is highly related to SLA. Scholar such as Lorido-Botran et.al [24] have described that there are five different types of scaling techniques:

- Static Threshold-based policies
- Reinforcement Learning
- Queuing Theory

- Control Theory
- Time Series Analysis

2.2.1 CPU Consumption and Power Relationship

In a data center it is difficult to conclude that CPU consumption and power usage are proportional. Since all processors (for example CPU,cooling fan and others) have their own energy consumption pattern. Studies show that servers need up to 70% of their maximum power consumption even at their low utilization level[8][13]. Thus, It can be concluded that the greater the CPU utilization the greater the power consumption. It was experimentally shown that the power consumption and CPU utilization have linear relationship by [19][10].

2.2.2 Dynamic Voltage and Frequency Scaling

Dynamic Voltage and Frequency scaling (DVFS) is a power management technique by modern processors to achieve energy efficiency. It is first utilized for mobiles computing but later on it is utilized for energy consumption in cluster of computers. It dynamically scales the supply voltage level of the CPU so as to provide circuit speed to process the system workload while meeting total computation time, which in effect reduce energy wastage[5]. As stated in chapter 1 under section 1.5 (The goal of the thesis), one of the important achievements of consolidating VMs is reducing the energy wastage by shutting down the idle resource (server) mainly by using the technique under discussion.

2.3 Data Center

Data center can be called a house of computer systems and components that make up organization's main body. In generailized term, a data center *"is a centralized repository, either physical or vitural, for the storage, management, and dissemintaion of data and information organized around a aprticulay body of knowledge."*Margareth Rouse (whatis.com) accessed 14.04.2015

2.4 Cloud Data Center

As cloud computing is the industry term for delivering hosted services over a network or the internet. It treats computing as a service rather than a product, enabling users to access and share a wide variety of applications, data, and resources through an interface such as their web browser[17]. Compared to traditional data center, cloud data center has decreased costs of infrastructure, flexibility to workload than tradition data center and many other more.

2.5 Covariance

Covariance is a measure of how two variables act together. It has been observed that those VMs that have covariance greater than 0 as high-variance and those with less than 0 are considered low-variance.

2.6 Correlation

Correlation is a normailized covariance. Figure 3.3 shows the correlation between VMs in the data set. Many studies have shown that consolidating based on correlation will help for efficient consolidation. Correlation standardizes the measure of interdependence between two variables and, consequently, tells how closely the two variables change. The correlation coefficient, will always take on a value between 1 and – 1. if it is close to -1 it means the correlation is strongly negatively correlated, i.e if one variable increases the other variable decreases. If it is 0 then it means there is no relationship between the variables. If it is larger than one then it means they are positively correlated, i.e as one variable increase the other variable also increases. A good candidate for VM consolidation is then negatively correlated VMs.

2.7 Related works

Similar works has been conducted by many research communities on dynamic VM consolidation, and the focus of many of them were to reduce resource consumption in data center, mainly power consumption. Cost reduction regarding power consumption and maximizing the server resource utilization were basically done by increasing packing efficiency of VMs and minimizing the number of used servers. In order to achieve the efficiency of VM consolidation different researchers approached it in so many different ways.

The authors of [14] proposes allocation and selection policy for the dynamic virtual machine (VM) consolidation in virtualized data centers to reduce energy consumption and SLA violation. Mean and standard deviation of CPU utilization for VMs were used to determine if the hosts are overloaded or not. Besides, they used the positive maximum correlation coefficient to select VMs from those overloading hosts for migration. Similar to our work mean are used to determine if the PM machines are overloaded, but our work uses variance and covariance instead of standard deviation in addition to mean. In addition the proposed allocation and selection policy uses probability to decide whether to put the VM on the PM.

Beloglazov and Buyya et al. in their works [11] [13]have implemented an energy-aware resource allocation heuristic for VMs consolidation. They proposed fixed threshold to migrate VMs on[11] and variable threshold in[13] in addition to new VM allocation and selection policies. The authors of [26] propose a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned together, based on an estimate of their aggregate capacity needs. Their work looks aw-some in reducing SLA violation, but complicated.

Bin packing algorithm was used by[4] assuming the resource usage will remain constant, which seems similar to what we have applied in our algorithm. They also characterize the resource usage dynamics by principal components, and propose to place VMs using variance reduction.

Chapter 3

Methodology

This chapter describes the experimental design, the set up and implementation part of the approach to pursue the goal of the thesis. Moreover, how to apply the designed algorithms will be discussed under this chapter. Different approaches to tackle the problem statement of the paper will be discussed here. Generally,explanation of how and why the approaches under discussion are chosen will be discussed under this chapter.

3.1 Objective of the experiment

The objectives of the experiment as mentioned briefly under the introduction part of this thesis is exploring how to improve VM consolidation by accessing time dependent behaviour of VMs and multiplex the VMs to reduce the total operational cost of a data center. The introduction of efficient algorithm that dynamically consolidate VMs with out violating SLA in a data center is of a focus. As discussed in previous chapter, reduction in operational cost of a data center includes:

- Reduction of energy consumption in the data center
- Effective utilization of physical machine
- Computational costs
- Network devices costs
- Environment costs(because of carbon emission)
- Cooling costs and the likes.

3.2 System Architecture

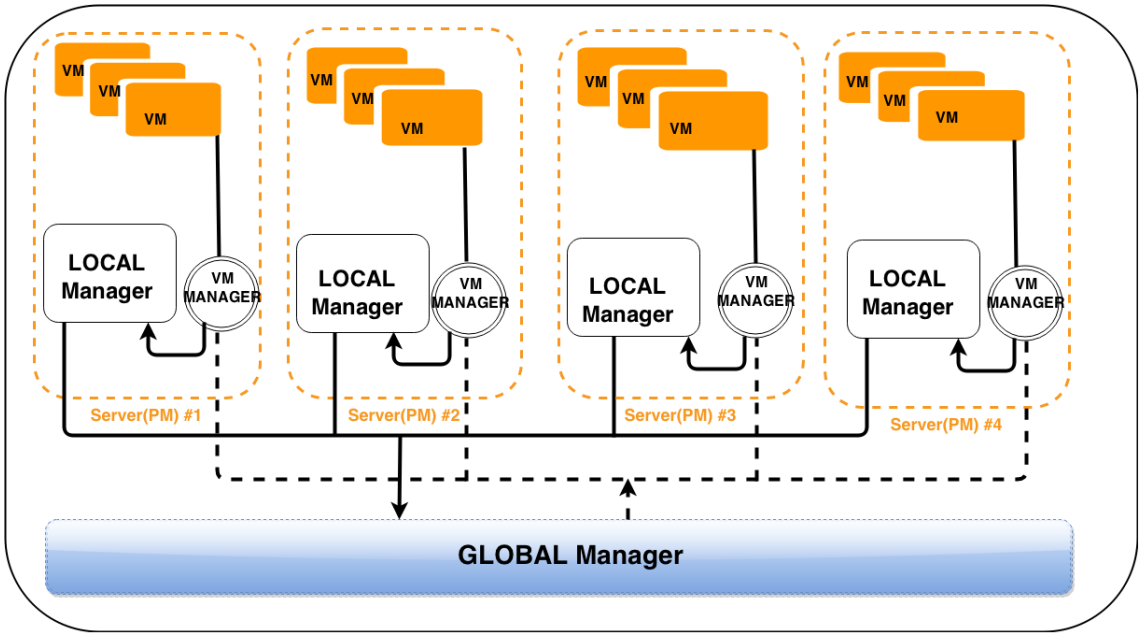


Figure 3.1: Architecture of the System Model

3.2.1 Description of the System architecture

It is assumed that a data center consists of homogeneous physical machine capable of executing heterogeneous VMs. Moreover, the following components of data center infrastructure are of significant value in consolidation.

- ***Local Manager:-***As it depicted on the figure of the system model above, local manager is found on physical machines. It manages CPU utilization and detects if the host (physical machine) is underloaded or overloaded and sends the information to global manager where the consolidation algorithm resides.
- ***Global Manager:-***it pulls the status of the physical machine and migrate the VM from the overloaded physical machine (Physical machine). In addition placement actions are also ordered by global manager. It gets the status information from local manager and communicates also with VM manager.
- ***VM Manger:-***basically VM manager is hyper-visor and it converts the order sent,i.e. migration and placement, from global manager to action.

3.2.2 Assumptions

- System Assumption: A data center with large number of Homogeneous physical machines.

- Resource assumption: CPU utilization demand of VM for the first time are assumed to be 100%
- Approach/algorithm: VM consolidation is considered as analogues to bin packing, which is an NP-hard problem. Therefore the algorithm used was heuristic greedy FFD algorithm that helps to reduce the number of physical machines utilized and SLA violation.
- Connection: It is assumed that all infrastructures are interconnected.
- Distribution: All Virtual machines (VMs) are considered as independent variable and follow normal distribution in stochastic approaches.
- Capacity of physical machine: in this paper it is assumed that capacity means the average amount of instructions that a physical machine can compute and or the average amount of resource demand of a VM. Though it is measured in hertz (hz) we simple put it as capacity.

3.2.3 Tools

- Lenovo: Intel Core i7-3537U CPU @2.00 GHz 2.5 GHZ
  RAM: 8GB
  OS: ubuntu
- R-studio: For ploting,
- Python:for scripting and simulation.
- Numpy: a Python library to compute all the necessary computational measures,such as mean, standard deviation,variance, covariance ...
- text editors: Nano,Latex

3.3 Data

The data that is going to be used in this work is real world data collected from PlanetLab presented by CoMon project available at https://github.com/beloglazov/planetlab-workload-traces as mentioned in introduction part. Special characteristics of the data is that it contains only the CPU utilization of each VM.Hence, CPU utilization by more than thousand VMs from physical machines spread over more than 500 places around the world with sampling interval of 5 minutes in a selected days in March and April 2011 are considered as an input. Totally, data for 3596 different VMs that are spread over 10 days as shown in the table below are collected. Out of the 10 days, data that are available in the first 8 days was chosen for learning their behaviour and the rest two days were left for prediction purpose. Thus, the designed model will be checked against these two days' data. The distribution overview of the workload is shown in table below.

Days	Date	Nr.of Vms
Day1	2011/03/03	1052
Day2	2011/03/06	898
Day3	2011/03/09	1061
Day4	2011/03/22	1516
Day5	2011/03/25	1078
Day6	2011/04.03	1468
Day7	2011/04/09	1358
Day8	2011/04/11	1233
Day9	2011/04/12	1054
Day10	2011/04/20	1033

Table 3.1: PlanetLab workload-traces collected in March and April 2011

3.3.1 Characteristics of the Data-set

The process of deducting properties of an underlying distribution by analysing the data is called Statistical Inference. Some distributions approximately represent the behaviors of a population. Statistical inferences use assumption to model statistical models. The normal distribution is a model identified by the mean and the variance. After selecting the type of distribution and model for the population, parameters for the model will be estimated to decide whether it fits the chosen model. This shows how well the model reflects the data set. Behaviours of statistical models are controlled by parameters. Thus, they are the important components in deciding the characteristics of the distribution. The parameters that are used in this project are mean, variance, and covariance. Since cloud VMs are of various types, actively running VMs show dynamic resource demands during run-time. As described above, the VMs that are collected by PlanetLab CoMon project shows such dynamic behaviour. Thus, their behaviour of CPU utilization will be used to perform workload prediction and estimation.

In order to start with the project, we selected a data-set of VMs out of collected data. The data-set consists of VMs that are mostly available or have the highest number of observation from day1 to day8 as a sample. Besides, the VMs that are selected for sample are cross-checked if they have observation in day9 and day10. A python script for accomplishing this is prepared and can be found under Appendix section (see Appendix A). A data set of 286 VMs out of the 3596 VMs in the data, are selected based on the aforementioned criteria.

After a sample data set is decided, the mean, variance and covariance of all VMs in a data set were computed. These parameters were all computed using the script mentioned. The out of the parameters were saved in to a file for later usage.

3.4 Experimental Setup

As it is mentioned earlier, our target system is IaaS (Infrastructue as a Service) and the system architecture is as shown in figure 3.1 above. it is difficult to conduct experiments with real infrastructure, however the importance of evaluating the approaches outlined with their corresponding algorithms on virtualized data center is huge. The cost and time frame allowed to conduct the experiment are some of various factors that are hinderance to experiment it with real infrastructure. Hence, simulation is the best alternative at hand to mimic a data center.

The following procedures were followed to obtain results. A script that collects the data set will be run first. The script computes, all the necessary inputs for bin-packing (i.e. mean, variance, and covariance. Next, the script that applies first fit decreasing bin-packing to the selected data will be run. In the scripts, physical machines are considered as bins and VMs are considered as items that are going to be place in the bin (see Appendix B, C, D for more). The result from first fit decreasing bin-packing algorithm of all approaches will be saved to a file. Each file contains the number of physical machines utilized to pack (consolidate) the VMs and also which VMs are packed on which physical machine. These generated files from the script run include the different scenarios used in the two last approaches (Variance and Covariance).Thus, there will be 14 files all together for the three approaches, 2 files for only mean-based (1 for capacity 120 and the other for capacity 140), 6 files for variance-based (3 for each of capacity 120 and 140 with scenarios (α=0.05, 0.5, 0.95), and 6 files for covariance-based. Moreover, a script that collects the total number of SLA violation will be run at the end. This script collects those VMs that has probability of exceeding the agreed up on service level agreement (SLA).

The result obtained will be shown in the next chapter and evaluation and analysis will be given in the corresponding chapter.

3.4.1 Approaches for VM Consolidation

In order to achieve our goal, we conducted three different approaches.

These Approaches are:-

- Deterministic Approach
- Stochastic approach I
- Stochastic approach II

Deterministic Approach

In deterministic approach we consider only mean to consolidate the VMs. Mean measures the central tendency of a probability distribution or of a random variable in a distribution. Such parameters are import to consider when we think of consolidation. Under this category the mean values of the CPU utilization of each VMs in a data set were taken. By mean value we

mean, first we compute the average of utilization in time series of 24 hours (i.e. [a VM has 288 entries]series of values in a day) for individual VM in their corresponding time slots in all days, this is computed using average() function in the main script (see Appendix A). Second, total average of all the averages for each VM is then computed to get the mean of a VM (see the allaverages() function of the main script, Appendix A, for detail information). The following table shows a sample example

time seies	day1	day2	average of day1, day2
1	3	6	4.5
2	8	9	8.5
3	2	3	2.5
4	5	6	5.5
5	8	9	8.5
.	.	.	.
.	.	.	.
.	.	.	.
288	4	6	5
-	-	Mean	**5.9**

Table 3.2: Table showing how mean of a VM is calculated

Thus, the mean of each VM is calculated as an example above when it comes to our data set. Hence, the normailzed will be

Generally, mean of individual VMs in our data-set of 286 VMs will be as follows

$$\mu_i = \frac{1}{\mathcal{D}^i * 288} \sum_{j=1}^{288} \sum_{d \in \mathcal{D}^i} X_{ijd} \tag{3.1}$$

where, X_{ijd} is observations for VM_i at time stamp j and day d and $\mathcal{D}^i$ is days with observation for VM_i

Suppose μ_1,...,μ_n are on a physical machine and we are considering to put μ_{n+1} on the physical machine

if μ_1+...+ μ_n + $\mu_{n+1} \leq cap$ **then**
    Put VM on the physical machine
else
    put μ_{n+1} on an other physical machine or start a new physical machine
end

where, μ_1 =mean of VM1 and cap=the capacity of the physical machine(physical machine).
From the above inequality it shows that if the sum of the means of the VMs are greater than the capacity of the physical machine , then do not put any more VM on that physical machine. The following pseudo code shows that.

Algorithm 1 Deterministic approach algorithm

Result: number of physical machines used to consolidate
cap=Physcial machine capacity
physical machine=[]
physical machine.append(VM(cap,[])) to start new physical machine
sort=sort the vms in decreasing load order

```
for elements in sort do
    if element > cap then
        exit //the VM can not be placed on any physical machine
    else
        for element in physical machine do
            if sum of elements in physical machine < cap and > element then
                Add element
            else
                if physical machine is full then
                    start new physical machine
                else
                end
            end
        end
    end
end
```

See the script under Appendix B for more.
The result obtained will be discussed under result chapter.

Stochastic Approach I

Variance shows how spread a data are around a mean. Under this model, mean and variance of the individual VMs are used. The procedure to compute variance is calculating the difference of every element of a set from a mean and average it. The advantage of using variance in VM consolidation is that, if for example two VMs have the same mean at their peak and the same variance, then the probability of exceeding the capacity is higher than those with the same mean but opposite variances. Hence, VMs with opposite or different variance will be a good candidate for consolidation in our case.

As stated earlier in this chapter, all VMs are assumed to be independent in this strategy. Thus, if for example a VM_1 has μ_1 and variance σ_1^2 and VM_2 has μ_2 and σ_2^2 then, the mean is equal to the sum of the means of the VMs and variance the square root of sum of the variances of VMs, in other words the mean of a physical machine is the sum of the means of the the two VMs already on the physical machine plus the mean of the new VM to be added and the summation of the variances of the two VMs plus the variance of the new VM to be added. Hence, the probability equation(3.2 and 3.3) will be checked against the threshold to put the new VM on the current physical machine.

Following this, it can be statistically put as follows. Suppose we have X_1, ...,X_n on a physical machine and consider to put X_{n+1}, then

$$\begin{pmatrix} X_1 \\ + \\ X_2 \\ + \\ X_3 \\ + \\ . \\ . \\ . \\ + \\ X_{n+1} \end{pmatrix} \sim N \left[\begin{pmatrix} \mu_1 \\ + \\ \mu_2 \\ + \\ \mu_3 \\ + \\ . \\ . \\ . \\ + \\ \mu_{n+1} \end{pmatrix}, \sqrt{(\ \sigma^2{}_1 + \ \ \sigma_2^2 + \ \ \sigma_3^2 + ... + \sigma_{n+1}^2 \ )} \right]$$

Thus,

$$\textbf{if} Pr(X_1 + X_2 + X_3 + ... + X_{n+1} \geq cap) \geq \alpha \textbf{then}$$

| start new physical machine and place X_{n+1} on it

end

(3.2)

$$\textbf{if} Pr(X_1 + X_2 + X_3 + ... + X_{n+1} \geq cap) \leq \alpha \textbf{then}$$

```
|   place X_{n+1} with X_1, ..., X_n
    end
(3.3)
```

where, Pr=Probability, α = probability for not exceeding the limit of the capacity of the physical machine. The following is the psudo code (Algorithm 2) for the stochastic bin packing algorithm.

Algorithm 2 Stochastic approach I algorithm

```
Result: Number of physical machine used for the consolidation
physical machine=[] start new physical machine
sort=sort the vms in decreasing load order

for element in sort do
    if element > cap then
        exit
    else
        for element in physical machine do
            calc= calculate the probability based on equation 3.2 and 3.3
        end
        if calc < threshold then
            add element to the physical machine
        else
            if physical machine is full then
                start new physical machine
            else
            end
        end
    end
end
```

Stochastic Approach II

Covariance is a measure of how much two variables change together. That is the degree to which two variables are linearly associated. Measuring how much two VMs change together will help in consolidating VMs.

$$COV_{xy} = \frac{1}{N}(\sum X_{iYi} - \sum X_i \sum Y_i) \tag{3.4}$$

Covariance for the data set is computed using the same script (see Appendix A) and the same computational procedure was followed to get the sum of all the means and covariances of a VM . The important point to be noticed here is that, VMs that tend to have high load at the same time (covariance greater than 0) will not be placed together.

Let us assume that the physical machine have $X_1, X_2, ..., X_n$, and consider to place X_{n+1} on it, then we have the probability distribution of the total load for the X_{n+1} VM as shown below

$$\begin{pmatrix} X_1 \\ + \\ X_2 \\ + \\ X_3 \\ + \\ . \\ . \\ . \\ + \\ X_{n+1} \end{pmatrix} \sim N \left[\begin{pmatrix} \mu_1 \\ + \\ \mu_2 \\ + \\ \mu_3 \\ + \\ . \\ . \\ . \\ + \\ \mu_{n+1} \end{pmatrix}, \sqrt{\left(\sum_{i=1}^{n+1} \sum_{j=1}^{n+1} Cov_{ij} \right)} \right]$$

where, Cov_{ij}= Covariance of VM_i and VM_j Hence,

if $Pr(\sum X_i > cap) > \alpha$ **then**

| start new physical machine and add X_n

end

(3.5)

if $Pr(\sum X_i > cap) < \alpha$ **then**

| Place X_{n+1} on current physical machine

end

(3.6)

Algorithm 3 Mean and Covariance based stochastic bin packing algorithm

Result: Number of physical machine used for the consolidation
physical machine=[]
physical machine.append(VM(cap,[])) to start new physical machine
sort=sort the vms in decreasing load order

```
for elements in sort do
    if element > cap then
        exit
    else
        for element in physical machine do
            calc= calculate the probability based on equation 3.5 and 3.6
        end
        if calc < treshold then
            Add element to physical machine
        else
            if the physical machine is full then
                start new physical machine
            else
            end
        end
    end
end
```

3.5 Policies for SLA

To keep truck of the agreed up on SLA our experiment implements a proactive rule-base scaling policies, that is based on the mean of the VMs which goes with the approaches selected above.

- *Overloaded detection Policy*:
  In order to maintain the SLA for the data center, we have applied two overloaded detection policies. First, the capacity of the VMs to be placed on the physical machine are checked if their CPU utilization demand exceeds that of the physical machine. Based on that, a limit is set to the capacity of the physical machine as a threshold and accomplish the bin-packing as described in equation 3.1 above. Second, the probability,as described in equations 3.2,3.3 and 3.5 and 3.6 above. Algorithm 2 and 3 above clearly shows this. Moreover, see Appendix A, B, C and D.

- *Selection Policy*
  After the hosts are detected and determined the designed algorithm will iteratively select the VMs and migrate them to other physical machine until the required SLA is met. The strategy to be used here is to select VMs based on either of the approaches discussed above.

- *Placement Policy:* In order to place VMs on the corresponding physical

machine, the well know heuristic First Fit Decreasing bin-packing technique was used. Bins are considered as physical machines to home the VMs and items are considered as VM to be guested on the physical machines. In all the three approaches discussed above the placement takes place based on the value the VMs have when the algorithm runs. That is the values of a VM is double checked both before placement and after placement. Thus, VM placement policy was introduced based on the capacity of the physical machine and demand of the VM.

Chapter 4

Results

In this section, the result obtained from the three approaches discussed in the previous chapter will be presented. As mentioned earlier, we conducted an experiment using three different approaches: deterministic approach, stochastic approach I and stochastic approach II. The techniques used for packing the VMs as mentioned was First Fit Decreasing bin-packing algorithm(see the algorithm at Appendix B,C and D). For the first approach, deterministic bin-packing based on mean of the VMs are used to consolidate the VMs. For the rest two approaches, stochastic bin packing algorithm was applied taking three different scenarios to compare the approaches.

4.1 Evaluation Metrics

In order to evaluate and compare the aforementioned approaches, the two metrics parameters that got attention based on the problem statement of this paper was the number of physical machines used to consolidate the VMs in the data set and percentage of SLA violation . The number of physical machines used, has also direct effect on the power consumption cost of a data center. One of the challenges of cloud data center is dynamic provisioning of resources. Due to fluctuation of CPU utilization of applications (VMs), there are often inefficiencies of resource provisioning resulting in performance degradation. This leads in effect leads to violation of service level agreement. In order to tackle this, the approaches under discussion are evaluated based on the total percentage SLA violation they incur. This is calculated as outlined in equation 4.1 below.

$$\frac{\textit{Total number of timestamps above the capacity for all physical machines}}{288(timestamps) * \textit{number of physical machine}} * 100 \tag{4.1}$$

Python script for computing the total SLA violation can be found under Appendix F.

The two important parameters compared in all of the three approaches were adjusting the capacity and thresholds. In effect, the number of PMs utilized and the total number of SLA violations to consolidate the

VMs will be evaulated. Thus, capacity was set to 120 and 140 for all approaches. Additional parameters, considered as threshold, for the last two approaches(i.e. stochastic approach I and II) was set to α=0.05, 0.5, and 0.95. Results obtained are shown on the table 4.1 and 4.2 for number of utilized PMs and 4.3 for total percentage of SLA violation.

4.2 Number of utilized PMs

Number of of used PMs, Mean-based		
Methods	Capacity=120	Capacity=140
Mean	24	20

Table 4.1: Number of PMs used, Out put from Deterministic approach

——————————— Capacity = 120—— | ————————capacity =140						
Methods	α =0.05	α =0.5	α =0.95	α =0.05	α =0.5	α =0.95
Variance	27	24	20	23	20	18
Covariance	27	24	18	23	20	16

Table 4.2: Number of PMs used by Stochastic Approach I and II

4.2.1 Number of VMs on each PM

The following graphs shows details of how many VMs a PM(physical machine) can accommodate using the three different approaches. For simplicity of reading, in figures from Figure 4.1 to 4.14, "value" on the y-axis indicates the average load of VMs, and "index" on the x-axis shows each PM with their corresponding VMs they accommodate.

4.2.2 Diterministic Approach VM Location

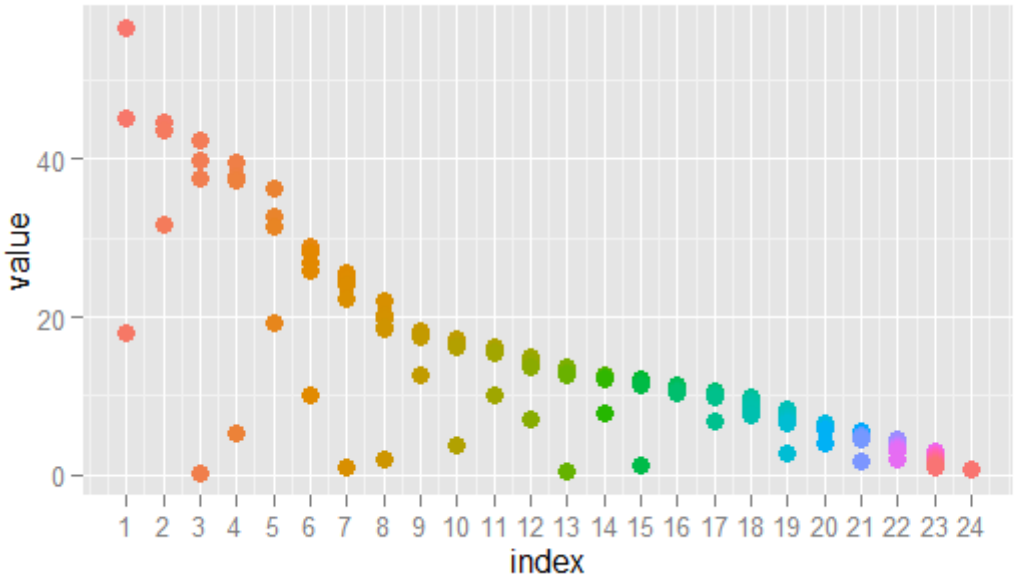


Figure 4.1: Number of VMs on each physical machine with capacity 120

4.2.3 Stochastic approach I VM location

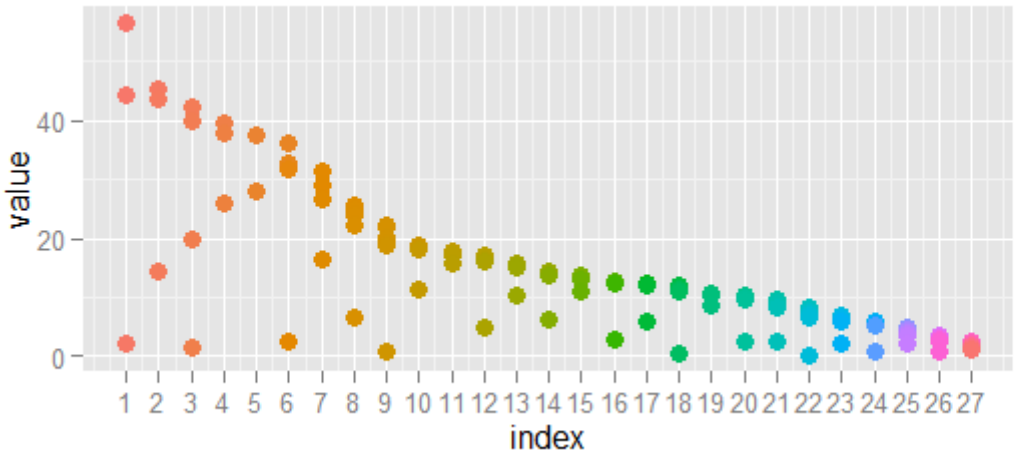


Figure 4.2: Number of VMs on each physical machine with capacity 120 and α=0.05

4.2.4 Stochastic Approach II VM location

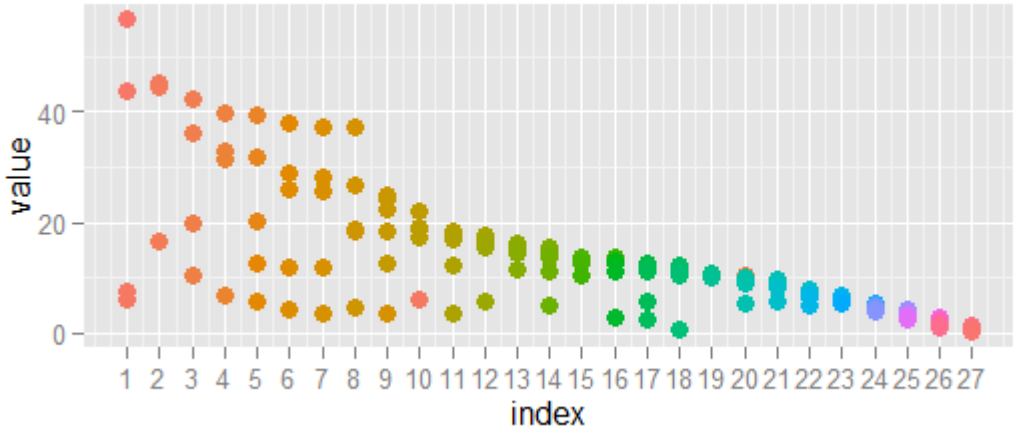


Figure 4.3: Number of VMs on each physical machine with capacity 120 and α=0.05

4.3 Total number of SLA Violations

The following table shows the total number of SLA violation in percentage in the learning days Day1 to Day8.

Total Number of SLA violations			
Methods	Times above Capacity	Number of PMs	Total violation(percentage)
Deterministic capacity=120	3171	24	45.8%
Deterministic capacity=140	2609	20	45.2%
Stochastic I capacity=120 and α=0.05	690	27	8.8%
Stochastic I capacity=120 and α=0.5	3169	24	45.8%
Stochastic I capacity=120 and α=0.95	4842	20	84%
Stochastic I capacity=140 and α=0.05	650	23	9.8%
Stochastic I capacity=140 and α=0.5	2609	20	45.2%
Stochastic I capacity=140 and α=0.95	4339	18	83.6%
Stochastic II capacity=120 and α=0.05	945	27	12%
Stochastic II capacity=120 and α=0.5	3169	24	45.8%
Stochastic II capacity=120 and α=0.95	4821	18	92.9%
Stochastic II capacity=140 and α=0.05	861	23	12.99%
Stochastic II capacity=140 and α=0.5	2609	20	45.2%
Stochastic II capacity=140 and α=0.95	4224	16	91.6%

Table 4.3: Total Number of SLA Violations in percentage

4.4 Description of Results

4.4.1 Observation on PMs utilization

As it can be seen from Table 4.1 and Table 4.2, though the threshold scenarios are the same, one can easily observe the following differences:-

- The number of PM used to accommodate the VMs in the data set shows significant differences.
- Deterministic approach with capacity 120 uses equal number of PMs with stochastic approach I with $\alpha = 0.5$. Similar observation are seen between deterministic approach with capacity 140 and stochastic approach with $\alpha = 0.95$.
- Types of VMs,i.e VM with average CPU load, on each PM are different for all approaches see Appendix F.
- Similar differences among the approaches and scenarios are observed on the graphs and tables.

Generally, in Table 4.1 and 4.2, stochastic approach I and II with capacity 120 and $\alpha = 0.05$ appears to use the maximum number of PMs to consolidate the VMs in the data set. Contrarily, the same approaches with capacity 120 and $\alpha = 0.95$ utilized the minimum number of VMs, i.e 20 and 18 respectively, to home the VMs.The deterministic approach in this regard shows neither maximum nor minimum utilization of the PMs. As α increases, naturally we can place more VMs on each PM. This is in accordance with table 4.2 and 4.3.

When the capacity for all approaches become 140, deterministic approach uses 20 and the stochastic approach I and II with $\alpha = 0.05$ uses the maximum number of PMs, i.e 23, to home the VMs. On the other side, stochastic approach I and II with $\alpha = 0.95$ used 18 and 16 PMs respectively. Again, the deterministic approach shows neither maximum nor minimum utilization of PMs with the capacity under discussion. Thus, it is observed that as α increases, more VMs can be place on the PM and the number of PM utilized decreases for capacity 140 too.

Observation on number of VMs on each PM

As it can be seen from the above figures, in Figure 4.1 the first and second PMs homed 3 VMs and the third PM has 4 VMs in deterministic approach. For stochastic approach I, figure 4.2 shows that the first, second and third PMs accomodated 3 VMs each and the fourth PM has 4 VMs on it. Further more, as Figure 4.3 shows using stochastic approach II the first PM homed 4 VMs and the second and third has 3 VMs each and so on. These shows that the capacity of the same VMs are welcomed differently for different approaches. The average CPU load of each VM is different in both scenarios(capacity 120 and 140). Thus, the capacity of the PM matters at-least to hold the type of VMs even though it is insignificant, and this in turn affects over all operational cost of the data center. For the rest of the result see Appendix F

4.4.2 Observation on SLA violation

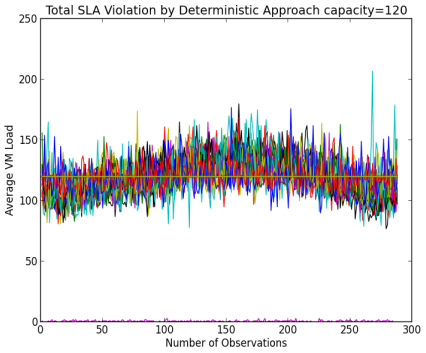
Table 4.3 shows the total number of SLA violation incurred in each approach. The maximum number of SLA violation incurred was by stochastic approach II with capacity 120 and $\alpha = 0.95$ which is ca. 92.9% followed by the same approach with capacity 140 and $\alpha = 0.95$(91.6%).

When capacity is 120 for all approaches the maximum SLA violation was done by stochastic approach II with $\alpha = 0.95$ and the minimum was done by stochastic approach I with $\alpha = 0.05$ which is ca. 8.8%.

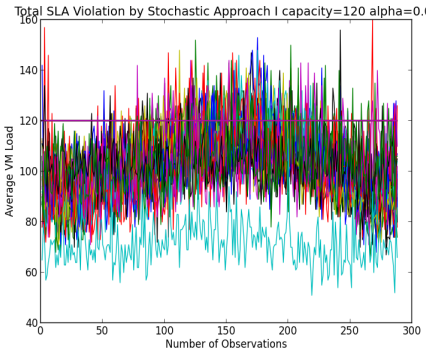
when capacity is 140 for all approaches the maximum SLA violation occurred was by stochastic approach II with $\alpha = 0.95$ which is 91.6% and the minimum violation was seen in stochastic approach I with $\alpha = 0.05$ which is 9.8%

The following graph, Figure 4.15 (a), (b), and (c) show the total violation of SLA in graphs by the three approaches: deterministic, stochastic I, and stochastic II, for Day1 to Day8 with capacity 120 respectively.

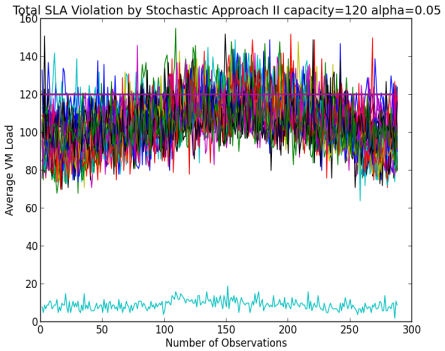
From Figure 4.15 and Figure 4.16, one can conclude that the trend of violating SLA shows the same pattern in both learning days (Day1 to Day8) and evaluation days (Day9 and Day10). This is because of the average of aggregate load values we use to compute load for VMs in different days for provisioning.



(a) Day1 to Day8 by Deterministic Approach



(b) Day1 to Day8 Stochastic approach I α



(c) SLA Violation by Stochastic Approach II with capacity 120 alpha=0.05

Figure 4.4: Total SLA violation for Day1 to Day8

4.5 Evaluation Days Result

4.5.1 Number of utilized PMs

Number of of used PMs, Deterministic Approach for Day9		
Methods	Capacity=120	Capacity=140
mean	24	20

Table 4.4: Number of PMs used by Deterministic approach for Day9 and Day10

Day9——————————— Capacity = 120——	————————-capacity =140					
Methods	α =0.05	α =0.5	α =0.95	α =0.05	α =0.5	α =0.95
Variance	27	24	20	23	20	7
Covariance	27	24	20	23	20	8

Table 4.5: Number of PMs used by Stochastic Approach I and II for day9 and day10

4.5.2 Total number of SLA violations

The following table shows the total number of SLA violation in percentage over evaluation days.

Total Number of SLA violations			
Methods	Times above Capacity	Number of PMs	Total violation(percentage)
Deterministic capacity=120	3173	24	45.8%
Deterministic capacity=140	2609	20	45.2%
Stochastic I capacity=120 and α=0.05	750	27	9.6%
Stochastic I capacity=120 and α=0.5	3169	24	45.8%
Stochastic I capacity=120 and α=0.95	4764	20	82.7%
Stochastic I capacity=140 and α=0.05	708	23	10.6%
Stochastic I capacity=140 and α=0.5	2609	20	48.2%
Stochastic I capacity=140 and α=0.95	4293	18	82.8%
Stochastic II capacity=120 and α=0.05	758	27	25.3%
Stochastic II capacity=120 and α=0.5	3169	20	45.8%
Stochastic II capacity=120 and α=0.95	4842	18	84%
Stochastic II capacity=140 and α=0.05	650	23	9.8%
Stochastic II capacity=140 and α=0.5	2609	20	45.2%
Stochastic II capacity=140 and α=0.95	4866	18	93.8%

Table 4.6: Total Number of SLA Violations in percentage in Day9 and Day10

4.6 Description of Evaluation Days Result

4.6.1 Observation on PM utilization

Deterministic Approach

The above tables, Table 4.5 and Table 4.6, show that the number of PMs utilized tend to go with the capacity. 24 PMs were used and 20 PMs with capacity 140 to consolidate VMs. This shows that number of PMs used decreases with increased capacity.

Stochastic Approach I and II

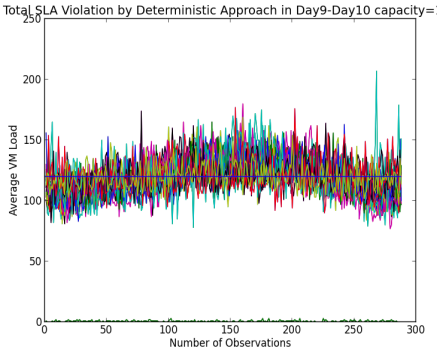
In both approaches, it shows that when capacity increase the number of PMs used decreases. But as α increases, both approaches show decrease in utilization of PMs. Special figure that is observed in the table is that, the stochastic approach I and stochastic approach II gives equal value in all the threshold values, α.

4.6.2 Observation on SLA violation

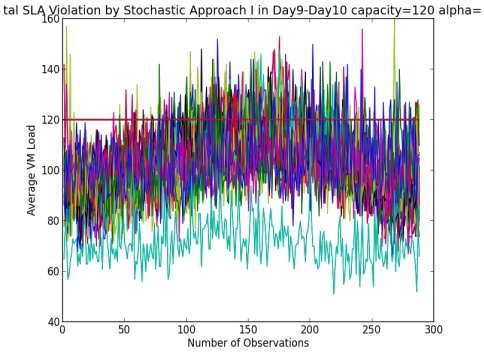
In deterministic approach total percentage violation of SLA seems to be constant or similar with increased capacity as observed from Table 4.6, but in stochastic approaches, it can be concluded that total percentage of SLA violation increases as α increases. In addition, stochastic approach II tends to use less PMs when α=0.95 in both capacities. For α=0.5 we observe no difference between the three approaches in violating total percentage of SLA.

Generally, it is observed that the number of PMs used are equal in all approaches when α=0.5. Furthermore, stochastic approach II tends to use minimum number of PMs and stochastic approach I and II show the maximum number of PM utilization. Moreover, stochastic approach I shows minimum total percentage of SLA violation and stochastic approach II showed maximum total percentage of SLA violation. In all cases, deterministic approach showed average in total percentage of SLA violation. Number of times above provisioned capacity increases with decrease in α, which is very natural, but in case of deterministic approach the result showed the opposite because of increase in capacity. Thus, when we compare Table 4.6 with Table 4.3 they follow the same pattern both in utilizing PMs and in total percentage of SLA violation.

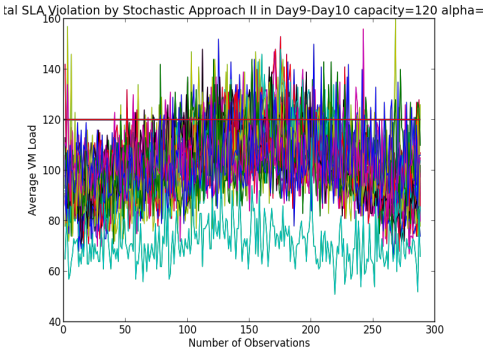
The following graph shows the total percentage of SLA violation by the three approaches with capacity 120. As depicted earlier, the graph follows the same pattern with that of learning days graph, Figure 4.15.



(a) Day9 and 10 by Deterministic Approach



(b) Day9 and Day10 Stochastic approach I α



(c) Day9 and Day10 Stochastic approach II α=0.05

Figure 4.5: Total SLA violation for Day9 and Day10

4.7 Covariance and Correlation matrix Result

Covariance and Correlation matrix for the VMs in the selected data set from all days is shown in Figure 4.6 and 4.7 respectively. This is just to show the ditribution of the data set. As it can be seen from Figure 4.6 below, there are very few observation with high-covariance.

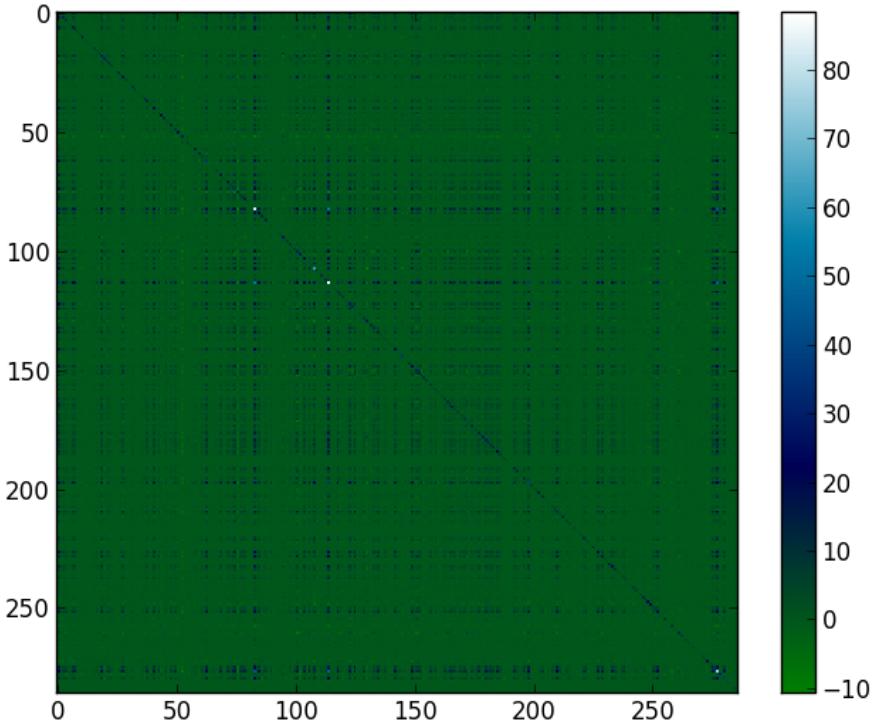


Figure 4.6: Covariance among the data-set

The correlation between pair of VMs in a data-set with time-stamp$(x_1, x_2, x_3, ..., x_n$ and $y_1, y_2, y_3, ..., y_n$were computed using Pearson Correlation Coefficient.

$$\rho_{xy} = \frac{COV_{xy}}{sd_x * sd_y} \tag{4.2}$$

where, sd_x and sd_y are standard deviation of x and y

Figure 4.7 shows that there are few VMs with both positively correlated and negatively correlated.

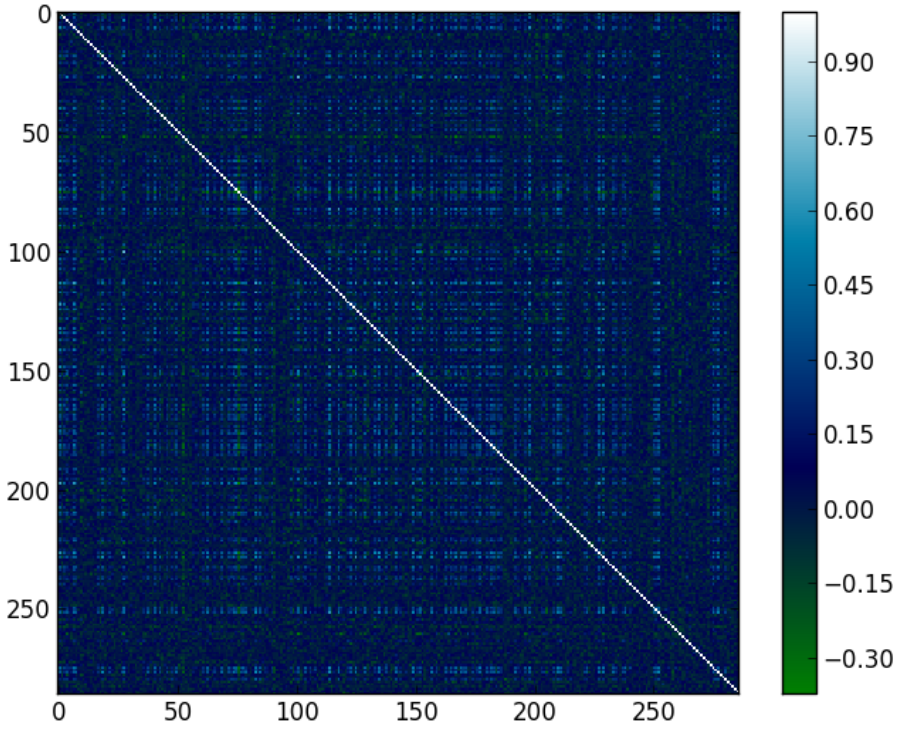


Figure 4.7: Correlation among the data-set

Chapter 5

Discussion

In this chapter, discussion and analysis of the results obtained will be presented in detail.

The goal of this paper is to explore and see how to reduce operational cost of a data center by consolidating VMs. The main target of consolidating VMs is to optimize the efficiency of resource utilization in the data center. Efficiently optimization of resources include the reduction of PM utilizatioin and SLA violation during consolidation. Thus, this paper focuses mainly on reducing the operational cost in light of reducing the total number of PMs used and the total number of SLA violations.

Several researches has been conducted on the subject of consolidating VMs in a data center to reduce the total operational cost. By reviewing related researches, we end up in designing our own algorithm to tackle the problems of a data center. In order to achieve this we first found ways of how to detect the overloading status of a PM. Secondly, techniques to migrate VMs from the overloaded hosts are selected. Thus, mean, variance and covariance were used to determine if the hosts are overloaded or not. Furthermore, first fit decreasing bin-packing technique was selected as mechanism to migrate VMs . Based on these three approaches, an algorithm was designed to conduct simulation of consolidating the VMs for our data set.

Results from the experiment of the simulation show that the deterministic approach showed similar results for violation of SLA both in learning days and evaluation days. In both cases, the only difference seen was on the number of PMs utilized when capacity changes. This by itself is very insignificant in reducing the SLA violation, though it showed positive effect. The same analysis holds true for the stochastic approaches, but with slight difference in percentage of SLA violation.

In stochastic approaches, increase in percentage of SLA violation goes with increase in α. But the number of PM utilization goes up inversely with α, that means when we increase α, minimum number of PMs are used. Though increasing the capacity of a PM helps in reducing the number of PMs used, it shows very little difference on SLA violation both in stochastic approach I and II. SLA violation increases with decrease in capacity in both deterministic and stochastic approaches.

As depicted on Table 4.3 in previous chapter, the same number of PMs are used by different approaches, for example deterministic approach with capacity 120 and stochastic approach with capacity 120 and α=0.5, use the same number of PMs and also the same percentage in SLA violation. Simultaneously, Table 4.6 shows the same result for deterministic approach and stochastic approach I and II with α=0.5.

Following that, we conclude that the effect of using covariance (stochastic approach II) in our case, for the selected data set is very low. This can be the result of too little dependencies between VM's real data.

There are different factors that can reduce the occurrence of SLA violation. Some of them are, limiting the provisioned capacity, which results in increasing the number of PMs and decreasing the number or percentage of SLA violation. For instance, we experimented this by provision only 90% of the capacity of the PM available. The following Table5.1 shows the result.

Total Number of SLA violations with 90 percent of the Capacity			
Methods	Times above Capacity	Number of PMs	violation (percentage)
Deterministic capacity=120	1274	26	17%
Deterministic capacity=140	1044	22	16.4%
Stochastic I capacity=120 and α=0.05	114	30	1.3%
Stochastic I capacity=140 and α=0.05	1373	26	18.8%
Stochastic II capacity=120 and α=0.05	141	30	1.63%
Stochastic II capacity=140 and α=0.05	1717	25	23.8%

Table 5.1: Total Number of SLA Violations after introduction of 90% of capacity

Thus, comparing the above Table 5.1 with Table 4.3 of previous chapter it can be concluded that if threshold is set to a provisioned capacity, it will end up in decreasing number of SLA violation by 83%, sacrificing additional 11% of the existing PMs.

In order to test the efficiency of the designed algorithm, the same data set of VMs are extracted from Day9 and Day10 of our obtained data. Then, the algorithm yielded the result on Table 4.6 that are found in previous chapter. The results depict that changing capacity have brought change in utilization of number of physical machines in deterministic approach, but showed almost no change in percentage violation of SLA the result as that of our learning days. In the other two approaches, stochastic approach I and II, total percentage of SLA violation increases with α. But, when we

look at total percentage of SLA violation as we increase capacity, it shows almost no change for α=0.5 similar to that of deterministic approach. The same analysis holds true for stochastic approach II. In general percentage of SLA violation goes inversely with capacity. Thus increased percentage of SLA violation shows decrease in number of PMs or vice versa.

5.1 Summary

From the experiment it is observed that there are trade-off between maintaining SLA and utilization of PM. The three approaches we selected also have shown different results leading us to conclude that one of the most challenging task in consolidating VMs is designing a system that balances between PM utilization and SLA. Thus, out of the three approaches implemented, it is observed that consolidation by stochastic approach II used minimum PM in both capacities and stochastic approach I showed minimum percentage of SLA violation in both capacities for VMs in learning days. Furthermore, it is observed that consolidation by stochastic approach I uses minimum PM in both capacities and at the same time have minimum percentage of SLA violation in both capacities for the evaluation days. Moreover, both learning days and evaluation days tend to use the same number of PMs in almost all of the approaches, except on the stochastic approach II with α=0.95. Thus, stochastic approach I will be selected for its minimum violation of SLA and stochastic approach II is selected for usage of minimum number of PMs.

Chapter 6

Conclusion and Future work

In cloud data centers one of the main goal of consolidating VMs is to optimize the efficiency of resource utilization and maintain SLA(service level agreement). One of the challenging tasks of cloud service providers is to provision required resources and at the same time keep SLA. In this paper, we experimented and presented three approaches to consolidate VMs with the aim of reducing the overall operational cost of a data center by minimizing the use of PMs and percentage of SLA violation. In order to do this, simulation on real-workload traces obtained from PlanetLab Workload-traces which was logged randomly every 5 minutes for 10 days in 2011 were selected. From the obtained data, VMs that have high observations in the learning days (Day1 to Day8) are selected as our target data set. Based on this analysis, a script that captures the workload of the VMs in a data set was developed. Besides, four similar but different scripts were developed to apply heuristic bin-packing script for deterministic approach, stochastic approach I and II, and SLA violation counter and plotter respectively (see the Appendix A, B, C, D and E for more).

The three approaches introduced have shown different results. Based on their results, we concluded that stochastic approach I withα=0.05 have shown good result for consolidating the VMs in our data set. Thus, we propose stochastic approach I approach for consolidation of VMs with minimum violation of SLA in both learning days and evaluation days. Stochastic approach II will be selected for minimum utilization of PMs for both learning days and evaluation days.

In all approaches, the results show that there are trade-off between percentage of SLA violation and utilization of PMs. It is already mentioned in earlier chapter that our system architecture is designed for IaaS (Infrastructure as a Service) which could be implemented for cloud service providers like open stack. Though the scope of this thesis is not up to that level, we propose that tailoring the problem between the service provider and consumer could be of reconciling solution when provisioning VMs. This assumption is based on the result we get from Table 5.1 of discussion chapter. The goal of this paper is exploring how to reduce the operational cost of a data center by consolidating VMs. One of the benefits of this work is reducing the number of used PMs and also minimize the total percentage

of SLA violation. Which in turn reduces the overall operational cost of a data center. From the experiment we discovered that it is difficult to guarantee 100% SLA in consolidation. But, further studies may result in more minimized SLA violation than we proposed.

Additional goal that can be pursued with the proposed methodologies mentioned is reduction of power consumption. Cost of power consumption in a data center is directly affected by number of physical machines utilized. This is because of linear relationship between CPU utilization and power consumption. Thus, we believe that we save cost of power consumption with the proposed methodologies, but incorporating calculation of power consumption could be one of the future works of such project. Furthermore, as mentioned earlier this work focused mainly only on the CPU utilization of VMs and the other constraints such as memory and bandwidth utilization are not included. Thus, incorporating them will be additional future work for such kind of project.

.1 Appendix A: Main Script

```
#!/usr/bin/env python
import os
import itertools
import numpy
import numpy as np
import numpy.ma as ma
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
from pylab import pcolor, show,colorbar,xticks,yticks,plot
from numpy import corrcoef, sum, log, arange
import math
from collections import Counter

path="workload path for the 10 days/"

################Reading Files####################
def readFile(f):
    list=[]
    file=open(f,"r")
    output=file.readlines()
    for l in output:
        list.append(int(l.strip()))
    return list

def listdirs(path):
    res=os.listdir(path)
    return res

#################collects VMs in a day bases######
def daysvms(path):
    full=[]
    counter=0
    day={}
    for root,sub,files in os.walk(path):
        if len(files)>0:
           r= root.split("/")
           day[r[1]]= files
    return day


####Helps to collect the unions and intersections of VMs in all
    days########
def uniques():


    d1= listdirs(path+"20110303")
    d2= listdirs(path+"20110306")
    d3= listdirs(path+"20110309")
    d4= listdirs(path+"20110322")
    d5= listdirs(path+"20110325")
    d6= listdirs(path+"20110403")
    d7= listdirs(path+"20110409")
    d8= listdirs(path+"20110411")
    d9= listdirs(path+"20110412")
    d10= listdirs(path+"20110420")
```

```
56
57      sets= set(d9) & set(d10)
58      uni=list(sets)
59
60      return uni
61 #####to count VMs that are available in many
62 #####of the days between Day1 and Day8
63 #####but should also be available in Day9 and Day10
64 def countvms(allvms,dayallvms):
65
66     dates = ["20110303", "20110306", "20110309", "20110322", "
       20110325", "20110403", "20110409", "20110411", "20110412", "
       20110420"]
67
68     list=[]
69
70     for vms in allvms:
71         counter=0
72         day=[]
73         for k,v in dayallvms.iteritems():
74             if vms in v:
75                 counter+=1
76                 day.append(k)
77         if counter >6:
78             if (dates[8] in day) & (dates[9] in day):
79                 list.append(vms)
80     return list
81
82 ####Reading the whole VM's file##########
83 def readRec(path):
84      full=[]
85      unique=countvms(allvms,dayallvms)
86      for root,sub,files in os.walk(path):
87          day={}
88          for file in files:
89               if file in unique:
90                   f=os.path.join(root,file)
91                   dayvms=readFile(f)
92                   day[file]=dayvms
93          full.append(day)
94      return  full[1:]
95
96 ####Collects together VMs in all days
97 def putvmsinsameday(data):
98      merged = {}
99      for d in data:
100         for k, v in d.items ():
101                 if k not in merged.keys():
102                     merged[k]=[]
103                 merged[k].append(v)
104     return merged
105
106
107 def vms_in_selected_days():
108      data=readRec(path)
109      d= putvmsinsameday(data)
110      return d
111
112 #### computes mean for vms in all days
```

```
113
114 def average():
115     dic=vms_in_selected_days()
116     res=[]
117
118     for key,value in dic.items():
119         value=[sum(col)/len(col) for col in zip(*value)]
120         res.append(value)
121          return res
122
123 #### computes average of the above calcuated mean of VMs
124 def allaverages():
125     y=average()
126     sum=(np.mean(y,axis=1))
127     tot=[]
128     for all in sum:
129         all=float(all)
130         tot.append(formater(all))
131
132     return tot
133
134 ### for formatting the digists to two decimal point
135 def formater(value):
136     f= "%.2f"%value
137     return float(f)
138
139
140 ### to compute covariance
141     dic=vms_in_selected_days()
142     res=[]
143     for key,value in dic.items():
144         #value=[formater(sum(np.cov(col))) for col in zip(*value)]
145         res.append(value)
146     yad=[]
147     for all in res:
148         value=[sum(np.cov(col)) for col in itertools.izip_longest
    (*all,fillvalue=0)]
149         yad.append(value)
150     vad=[]
151     for all in yad:
152         dav=sum(all)
153         vad.append(dav)
154 #         fint=sum(all)
155 #          yad.append(formater(fint))
156     k=[]
157     for n in vad:
158         j=formater(n)
159         k.append(j)
160     return k
161
162
163 ### Computes variance
164 def variance():
165
166     z=average()
167     stnd=np.var(z,axis=1)
168     devar=[]
169     for all in stnd:
170         all=float(all)
```

```
171             devar.append(formater(all))
172         return devar
173 ##########################################
174 ### Computation for Day9 and Day10 comes here####
175
176 def daysnine_ten():
177     #date= ["day4", "day5"]
178     date= ["20110412", "20110420"]
179     dayallvms= daysvms(path)
180     allvms= uniques()
181 #     list=[]
182     for all in date:
183             list=[]
184             for k,v in dayallvms.iteritems():
185                     if k in date :
186                         list.append(v)
187     new=[]
188     for all in list:
189         for ele in all:
190             new.append(ele)
191     new3=set(new) & set(countvms(allvms,dayallvms))
192     return new3
193
194 def daysonetoeight():
195
196 #     spec=countvms(allvms,dayallvms)
197
198     date= ["20110303", "20110306","20110309","20110322","20110325"
      ,"20110403","20110409","20110411"]
199 #     date= ["day1", "day2","day3"]
200     allvms= uniques()
201     dayallvms= daysvms(path)
202 #     list=[]
203     for all in date:
204             list=[]
205             for k,v in dayallvms.iteritems():
206                     if k in date :
207                         list.append(v)
208     new=[]
209     for all in list:
210         for ele in all:
211             new.append(ele)
212
213     new2=set(new) & set(countvms(allvms,dayallvms))
214
215     return new2
216
217 ##########################################
218
219
220 ###generate correlation matrix and plots it
221
222 def corrmat():
223     xp=average()
224     ph=np.corrcoef(xp)
225     return ph
226
227 def plothis():
228             fig=plt.figure()
```

```
229         ax=fig.add_subplot(1,1,1)
230         ax.set_aspect('equal')
231         plt.imshow(corrmat(),interpolation='nearest',cmap=plt.cm.
    ocean)
232         plt.colorbar()
233         y=plt.show()
234         return y
```

Listing 1: Main Script

.2 Appendix B: Bin Packing Script for Deterministic approach script

```
1
2 #!/usr/bin/env python
3
4 # First-fit Decreasing approximation algorithm for Bin Packing
    based on mean
5
6 import math
7 import sys
8 from time import time, clock
9 import finalscript as fin
10
11 class Bin:
12         """ For putting items in it"""
13         def __init__(self, capacity, contents=[]):
14                 self.capacity = capacity
15                 self.contents = contents
16         def add(self, x):
17                 self.contents.append(x)
18         def __repr__(self):
19                 return str(self.contents)
20 cap = 120
21 items = fin.allaverages()
22 bins = []
23 bins.append(Bin(cap, [])) # we need at least one bin to begin
24
25 items = sorted(items) # sort in descending order
26
27 for item in reversed(items): #iterate through the list backwards
28   # Add the item to the first bin that can hold it
29   # If no bin can hold it, make a new bin
30   if item > cap:
31     print "THE CAPACITY OF THE VM IS GREATER THAN THAT OF SERVER!"
32     sys.exit()
33   for vms in bins:
34
35         if vms.capacity - sum(vms.contents) >= item
36       vms.add(item)
37       break
38     if bins.index(vms) == len(bins) - 1:
39
40       bins.append(Bin(cap, []))
41
42 print "The consolodated VMs were", bins
```

Listing 2: Deterministic Approach Script

.3 Appendix C: Stochastic I approach bin packing script

```
1
2 #!/usr/bin/env python
3 # First-fit Decreasing approximation algorithm for Bin Packing
4 import numpy as np
5 import math
6 from math import *
7 import sys
8 import scipy
9 import finalscript as fin
10 from scipy import stats
11 class Bin:
12         """For putting items in it"""
13         def __init__(self, capacity, contents=[]):
14                 self.capacity = capacity
15                 self.contents = contents
16         def add(self, x):
17                 self.contents.append(x)
18         def __repr__(self):
19                 return str(self.contents)
20
21 def mean_variance():
22
23     x=fin.allaverages()
24     y=fin.variance()
25
26     counter=0
27     tuples=[]
28     while counter<len(x):
29         tuples.append((x[counter], y[counter]))
30         counter+=1
31     sort= sorted(tuples, key=lambda tup: tup[0])
32     return sort
33
34
35 cap =120
36 items=mean_variance()
37 bins = []
38 bins.append(Bin(cap, [])) # we need at least one bin to begin
39
40 f
41 for item in reversed(items): #iterate through the list backwards
42
43         if item[0] > cap:
44                 print "CAPACITY OF THE VM IS GREATER THAN THE
    CAPACITY OF THE SERVER! ABORTING"
45                 sys.exit()
46         for vm in bins:
47                 calc=1-stats.norm.cdf(cap,loc=(sum(vm.contents)+
    item[0]),scale=math.sqrt(sum(vm.contents)+item[1]))
48                 if calc <0.95:
49                     vm.add(item[0])
50
51                     break
52                 if bins.index(vm) == len(bins) - 1:
```

```
53
54                          bins.append(Bin(cap, []))
55
56
57 print "\n\nThe consolidation is as follows:\n\n", bins
58
59
60
```

Listing 3: Stochastic I script

.4 Appendix D: Stochastic Approach II bin packing Script

```
1
2 #!/usr/bin/env python
3  # First-fit Decreasing approximation algorithm for Bin Packing
4 import numpy as np
5 import math
6 from math import *
7 import sys
8 import scipy
9 import finalscript as fin
10 from scipy import stats
11 class Bin:
12         """For putting items in it"""
13         def __init__(self, capacity, contents=[]):
14                 self.capacity = capacity
15                 self.contents = contents
16         def add(self, x):
17                 self.contents.append(x)
18         def __repr__(self):
19                 return str(self.contents)
20
21 def mean_covariance():
22
23     x=fin.allaverages()
24     y=fin.covariance()
25
26     counter=0
27     tuples=[]
28     while counter<len(x):
29         tuples.append((x[counter], y[counter]))
30         counter+=1
31     sort= sorted(tuples, key=lambda tup: tup[0])
32     return sort
33
34
35 cap = 120
36 items=mean_covariance()
37 bins = []
38 bins.append(Bin(cap, [])) # we need at least one bin to begin
39
40 for item in reversed(items): #iterate through the list backwards
41
42         if item[0] > cap:
43                 print "CAPACITY OF THE VM IS GREATER THAN THE
    CAPACITY OF THE SERVER! "
44                 sys.exit()
45         for vm in bins:
46                 calc=1-stats.norm.cdf(cap*0.9,loc=sum(vm.contents)
    +item[0],scale=math.sqrt(abs(sum(vm.contents)+item[1])))
47                 if calc <0.05:
48                     vm.add(item[0])
49
50                     break
51                 if bins.index(vm) == len(bins) - 1:
52
```

```
53                      bins.append(Bin(cap, []))
54
55
56 print "\n\nThe consolidation is as follows:\n\n", bins
```

Listing 4: Stochastic Approach II script

.5 Appendix E: SLA Violatioin Counter and grapher

```
1
2  #!/usr/bin/env python
3  import os
4  import sys
5  import ast
6  import numpy as np
7  import matplotlib.pyplot as plt
8  import itertools
9  from itertools import *
10
11 sameall.txt=(allaverages:mean of vms)
12 alllist={}
13 with  open("sameall.txt","r") as f:
14         lines=f.readlines()
15         for line in lines:
16            l=line.split(':')
17            alllist[float(l[0].strip())]= eval(l[1].strip())
18
19 total=[]
20
21 for l in listofservers:
22     list=[]
23
24     for item in l:
25          list.append(alllist.get(item))
26     total.append(list)
27 sumia=[]
28 for all in total:
29     fa=[sum(i) for i in zip(*all)]
30     sumia.append(fa)
31
32
33 for all in sumia:
34         x=[x for x in range(1,289)]
35 #       y2=[140]*288
36         y3=[120]*288
37         plt.plot(x,all,label='vm')
38  #      plt.plot(x,y2,label='cap')
39         plt.plot(x,y3,label='cap2')
40         plt.xlabel("Number of Observations")
41         plt.ylabel("Average VM Load")
42         plt.title("Total SLA Violation by Stochastic Approach II
       capacity=120 alpha=0.05 ")
43 plt.savefig("/Users/yadassaa/Desktop/cov120-005.png")
44 plt.show()
45 plt.close()
46
47
48 yad=[]
49 for all in sumia:
50         for ele in all:
51                 if ele>140:
52                         yad.append(ele)
53
54
55
```

56

Listing 5: SLA Violation counter and grapher

.6 Appendix F: Miscellaneous Figures

.6.1 Number of VMs on each PM

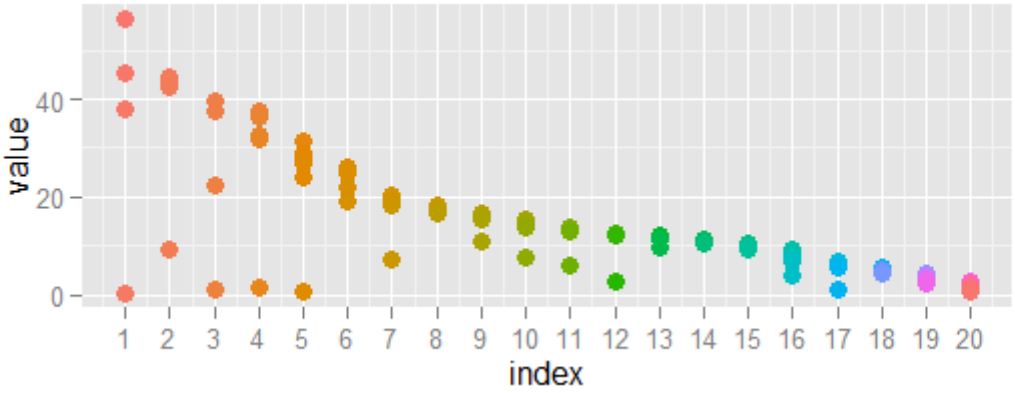


Figure .61: Number of VMs on each physical machine with capacity 140

.6.2 Stochastic approach I VM location

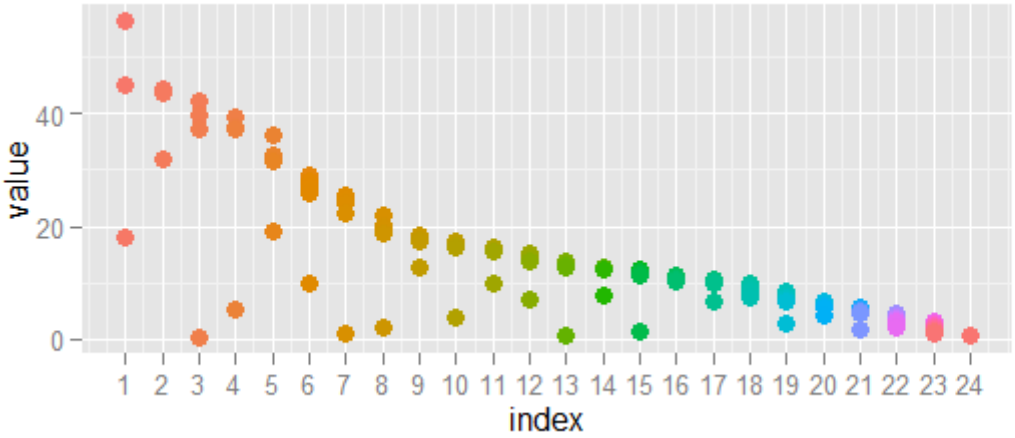


Figure .62: Number of VMs on each physical machine with capacity 120 and α=0.5

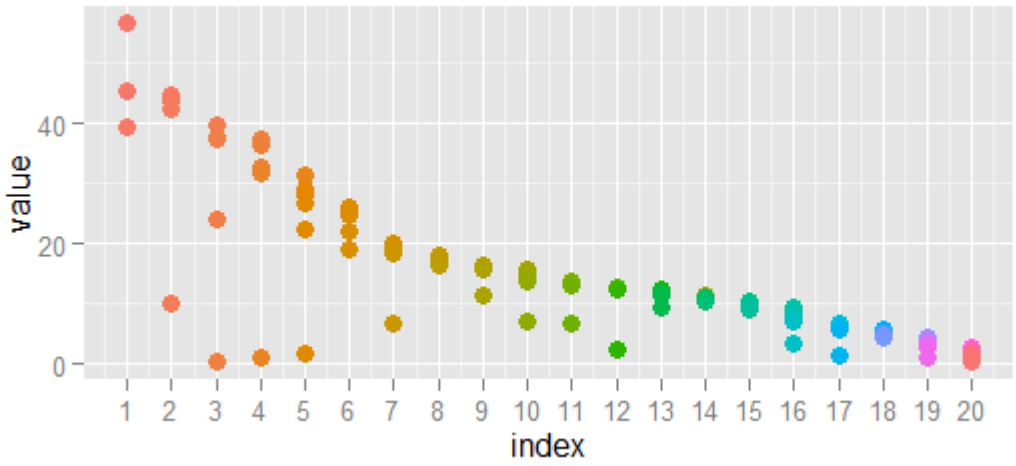


Figure .63: Number of VMs on each physical machine with capacity 120 and α=0.95

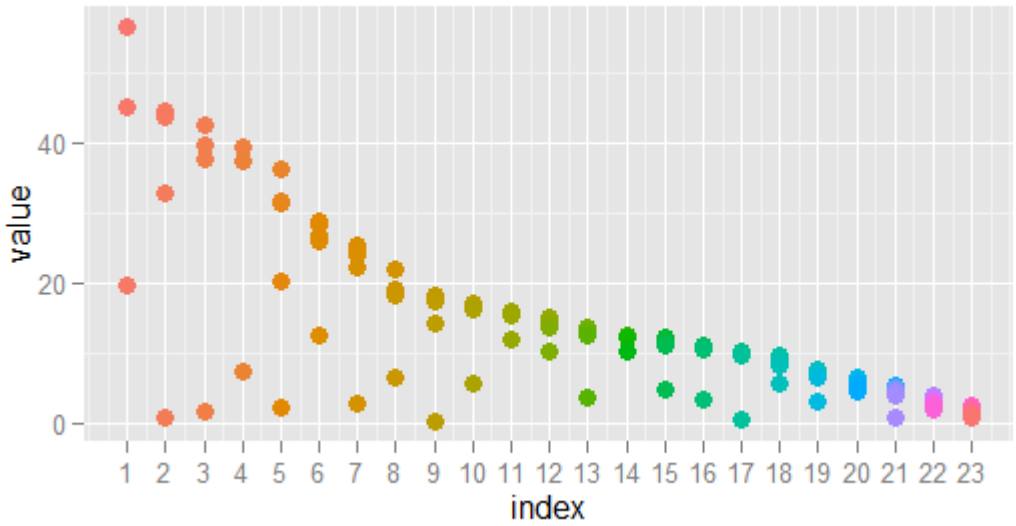


Figure .64: Number of VMs on each physical machine with capacity 140 and α=0.05

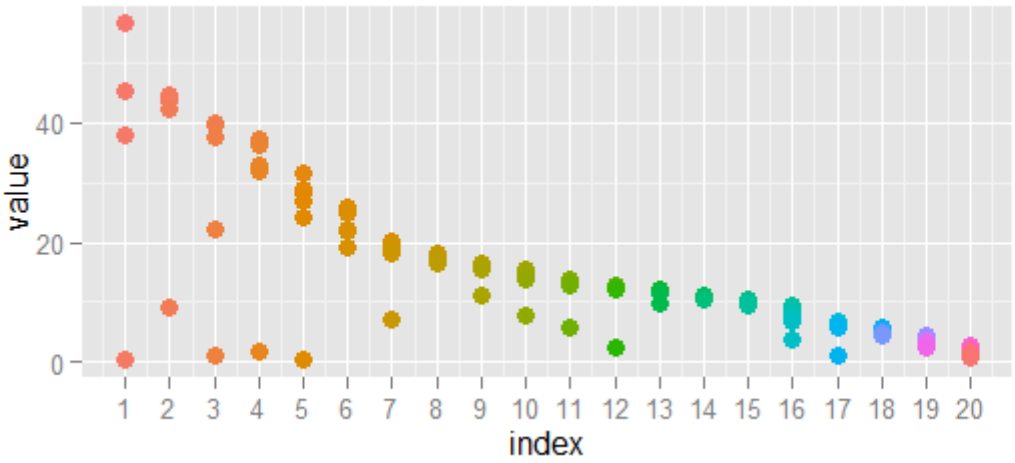


Figure .65: Number of VMs on each physical machine with capacity 140 and α=0.5

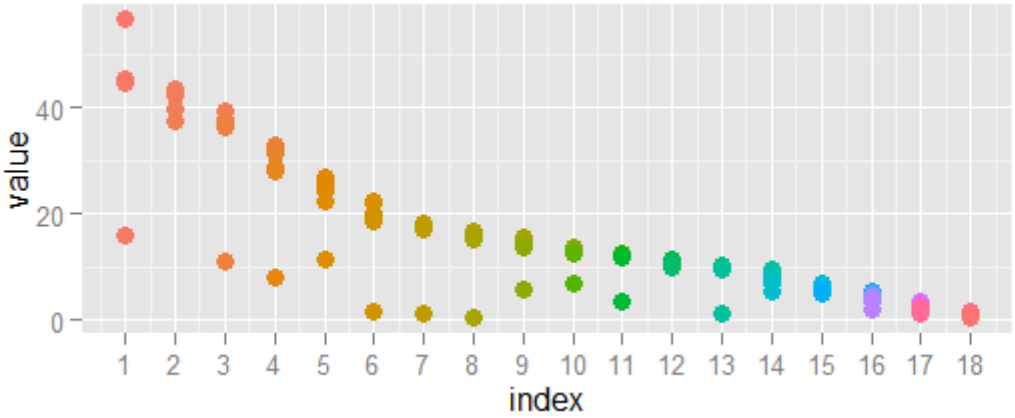


Figure .66: Number of VMs on each physical machine with capacity 140 and α=0.95

.6.3 Stochastic approach II VM location

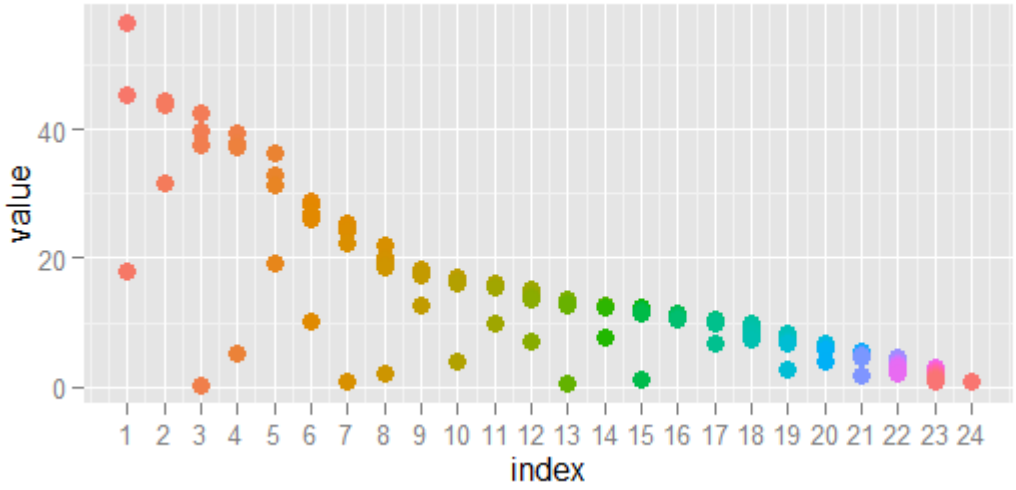


Figure .67: Number of VMs on each physical machine with capacity 120 and α=0.5

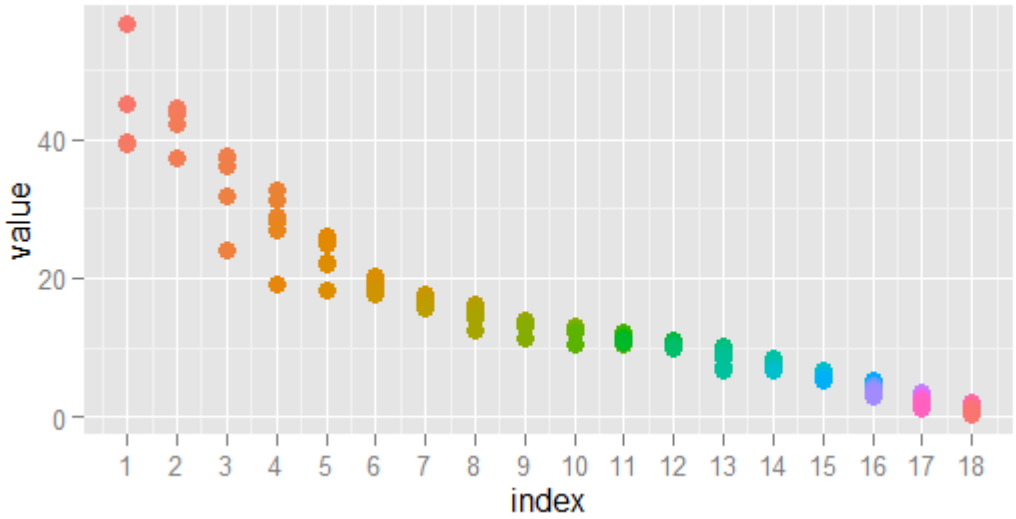


Figure .68: Number of VMs on each physical machine with capacity 120 and α=0.95

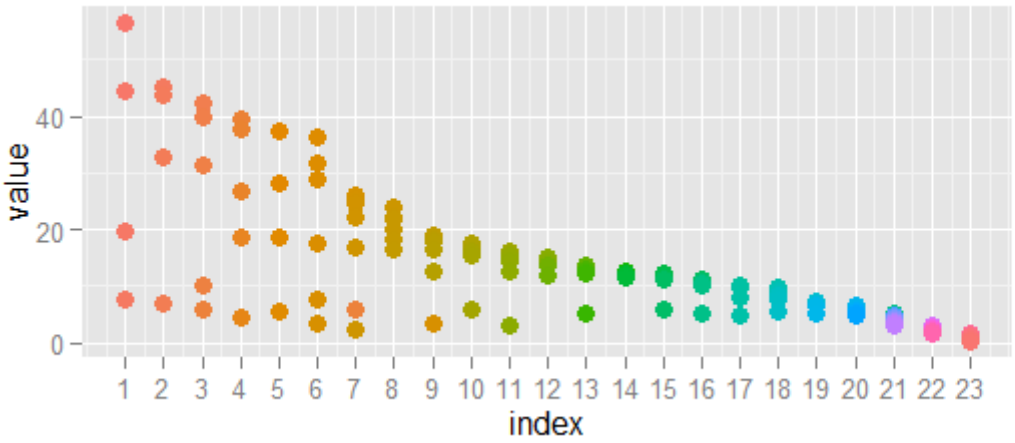


Figure .69: Number of VMs on each physical machine with capacity 140 and α=0.05

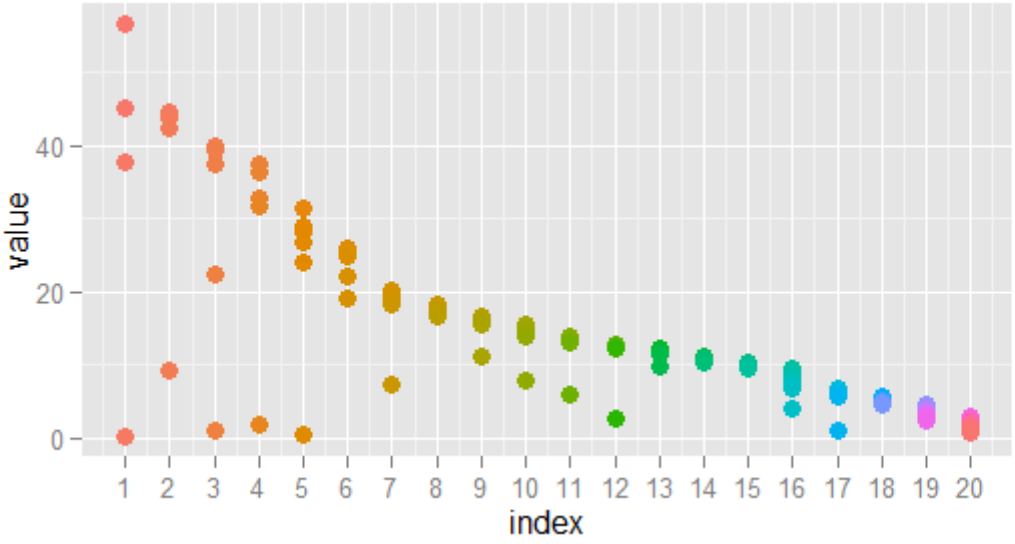


Figure .610: Number of VMs on each physical machine with capacity 140 and α=0.5

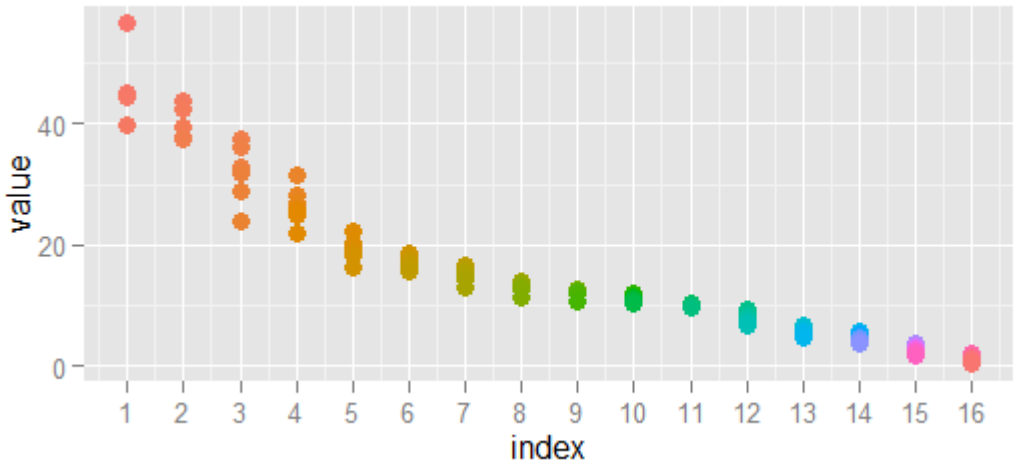


Figure .611: Number of VMs on each physical machine with capacity 140 and α=0.95

Bibliography

[1] Amany Abdelsamea et al. 'Virtual Machine Consolidation Challenges: A Review'. In: *International Journal of Innovation and Applied Studies* (2014), pp. 1504–1516.

[2] Hadi Goudarzi et al. 'SLA-based Optimization of Power and Migration Cost in Cloud Computing'. In: *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (2012), pp. 172–179.

[3] Heena Kaushar et al. 'omparison of SLA based Energy Efficient Dynamic Virtual Machine Consolidation Algorithms'. In: *International Journal of Computer Applications* 102 (16 2014), pp. 31–36.

[4] Jian Tan et al. 'Exploiting Resource Usage Patterns for Better Utilization Prediction'. In: *2011 31st International Conference on Distributed Computing Systems Workshops* 8 (4 2011).

[5] Kihwan Choi et al. 'Dynamic Voltage and Frequency Scaling based on Workload Decomposition'. In: *ACM* (2004), pp. 1–6.

[6] Kishaloy Halder et al. 'Risk Aware Provisioning and Resource Aggregation based Consolidation of Virtual Machines'. In: *2012 IEEE Fifth International Conference on Cloud Computing* (2012), pp. 598–605.

[7] Norman Bobroff et al. 'Dynamic Placement of Virtual Machines for Managing SLA Violations'. In: *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies* (2007), pp. 119–128.

[8] P. Getzi Jeba et al. 'LIVE VM MIGRATION TECHNIQUES IN CLOUD ENVIRONMENT – A SURVEY'. In: *Proceedings of 2013 IEEE Conference on Information and Communication Technologies* (2013), pp. 104–113.

[9] Rina Pangigrahy et al. 'Heuristics for Vector Bin Packing'. In: *Microsoft Research* (Unpublished 2011).

[10] Wen Chengjian et al. 'Dynamic Power Saving via Least-Square Self-Tuning Regulator in the Virtualized Computing Systems'. In: *The Editorial Office of JSSC Springer* (2012).

[11] Anton Beloglazov, Jemal Abawajy and Rajkumar Buyya. 'Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing'. In: *Future Generation Computer Systems* 28.5 (2012), pp. 755–768.

[12] Anton Beloglazov and Rajkumar Buyya. 'OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds'. In: *Concurrency and Computation:Practice and Experience* (2014).

[13] Anton Beloglazov and Rajkumar Buyya. 'Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers'. In: 24.13 (Sept. 2012), pp. 1397–1420.

[14] Zhibo Cao and Shoubin Dong. 'Dynamic VM Consolidation for Energy-Aware and SLA Violation Reduction in Cloud Computing'. In: *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2012 13th International Conference on*. Dec. 2012, pp. 363–369. DOI: 10.1109/PDCAT.2012.68.

[15] Cloudsim. *What is CloudSim*. Feb. 2015. URL: https://code.google.com/p/cloudsim/wiki/FAQ#1.%20_What_is_CloudSim_?_What_does_it_do_and_what_doesn't_it_d.

[16] Pierre Delforge. 'America's Data Centers Consuming and Wasting Growing Amounts of Energy'. In: *http://www.nrdc.org/energy/data-center-efficiency-assessment.asp* (2014).

[17] Eamonn. *What is the difference between IaaS,SaaS,PaaS?* Feb. 2015. URL: https://www.computenext.com/blog/when-to-use-saas-paas-and-iaas/.

[18] Peter mell etal. *The NIST Definition of Cloud Computing*. 2011. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[19] Xiaobo Fan, Wolf-Dietrich Weber and Luiz Andre Barroso. 'Power Provisioning for a Warehouse-sized Computer'. In: *Proceedings of the 34th Annual International Symposium on Computer Architecture*. ISCA '07. ACM, 2007, pp. 13–23.

[20] Bill Kleyman. *Data Center Optimization*. Jan. 2014. URL: http://www.datacenterknowledge.com/archives/2014/01/22/five-great-ways-optimize-data-center/.

[21] Bill Kleyman. *Understanding the Virtualization market*. 2015. URL: http://www.datacenterknowledge.com/archives/2012/08/01/hypervisor-101-a-look-hypervisor-market/.

[22] Jonathan G Koomey. 'Worldwide electricity used in data centers'. In: *Lawrence Berkeley National Laboratory, USA* (2008), pp. 1–9.

[23] Century Link. *Cloud Data Center Service Level Agreement*. 2015. URL: http://www.centurylinktechnology.com/en-gb/legal/sla/cloud-data-center.

[24] Tania Lorido-Botrán, José Miguel-Alonso and Jose Antonio Lozano. *Auto-scaling Techniques for Elastic Applications in Cloud Environments*. Research EHU-KAT-IK. Department of Computer Architecture and Technology, UPV/EHU, 2012.

[25] Zar Lwin and Thandar Thein. 'Correlation based VMs Placement Resource Provision'. In: *International Journal of Computer Science Information Technology (IJCSIT) Vol 5, No 1, February 2013* (2013), pp. 95–107.

[26] Xiaoqiao Meng et al. 'Efficient Resource Provisioning in Compute Clouds via VM Multiplexing'. In: *Proceedings of the 7th International Conference on Autonomic Computing*. ICAC '10. New York, NY, USA: ACM, 2010, pp. 11–20.

[27] Micrsoft. *About Virtual Machine Placement*. Feb. 2015. URL: https://technet.microsoft.com/en-us/library/bb740817.aspx.

[28] Margaret Rouse. *Virtualization*. Feb. 2015. URL: http://searchservervirtualization.techtarget.com/definition/virtualization.

[29] M. Sedaghat et al. 'Divide the Task, Multiply the Outcome: Cooperative VM Consolidation'. In: *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*. Dec. 2014, pp. 300–305. DOI: 10.1109/CloudCom.2014.81.

[30] *Virtualization*. Feb. 2015. URL: http://en.wikipedia.org/wiki/Virtualization.

[31] Wikipedia. *Bin Packing Problem*. March 2015. URL: http://en.wikipedia.org/wiki/Bin_packing_problem.

[32] Wikipedia. *Virtual Machine*. 2015. URL: http://en.wikipedia.org/wiki/Virtual_machine.

[33] A. Wolke and C. Pfeiffer. 'Improving Enterprise VM Consolidation with High-Dimensional Load Profiles'. In: *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. Mar. 2014, pp. 283–288.

[34] Xiaodong Zhang et al. 'SmartRelationship: A VM Relationship Detection Framework for Cloud Management'. In: *Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware*. INTERNETWARE 2014. ACM, 2014, pp. 72–75.